PARAMETER ESTIMATION IN MODELS GOVERNED BY ORDINARY DIFFERENTIAL EQUATIONS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By

SUHAS KUMAR

to the

DEPARTMENT OF CHEMICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JULY, 1984

CERTIFICATE

It is certified that the work entitled 'Parameter estimation in models governed by ordinary differential equations' has been carried out under my supervision and has not been submitted elsewhere for a degree.

Date: July 5, 1984.

(DR. M.S. RAO)
PROFESSOR and HEAD
DEPARTMENT OF CHEMICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR -208016

ACKNOWLEDGEMENT

My thanks, in all sincerity, are to Prof. M.S. Rao but for whose valuable guidance and advice it would not have been possible for me to complete this work successfully.

I would like to thank Mr. S. Singh, Department of Mathematics for his help and cooperation.

I am most grateful to my good friends, Srinivasan and Subramanyam who generously gave of their time to read the text for necessary corrections in the typed material.

Last, but not the least, thanks are to Mr. S.N. Pradhan who most admirably transformed the manuscript into a neat typewritten form.

(SUHAS KUMAR)
Author

CONTENTS

LIST OF TABLES

NOMENCLATURE

$\underline{X}$	n- dimensional state vector defined in Eq. (13)
$\underline{k}$, $\underline{b}$	p- dimensional parameter vectors defined in Eq. (13)
$\underline{Y}$	m- dimensional vector of responses defined in Eq.(14)
$\underline{Y}$	m-dimensional vector of predicted responses
t_r	value of independent variable, t, at r^{th} observation
S	objective function used in the minimization routines
$\underline{Q}_r$	mxm positive definite weighting matrix for r^{th} obs.
$\underline{Z}$	augmented (n+p) dimensional vector of state variables and parameters as given by Eq (17)
$\underline{g}$	augmented (n+p) dimensional vector of functions of state vector $\underline{X}$
C_i	concentrations of M components involved in the reaction scheme given by Eq.(19)
B_j & $R_{i,j}$	rate constants and concentration terms involved in the reaction scheme of Eq (19)
X_{ijk}	$\int_{t_o}^{t_i} R_{j,k}\, dt$
$V(\underline{b})$	variance-covariance matrix of the estimated parameters of order pxp
$\underline{y}$	n-dimensional state vector defined in Eq.(25)
$\underline{y}_{n+1}$	value of the state vector $\underline{y}$ at $(n+1)^{th}$ observation of time as predicted by the model
A, B, C & D	Runge Kutta-Gill constants defined in Eq.(27) also used as components in Eq(a)-(d) on pp. 62

h	stepsize in numerical integration schemes employed
$\underline{h}$	vector of functions of the state vectors predicted by the model defined in Eq.(14)
$\underline{\underline{A}}'$	nxm order matrix of concentration terms defined in Eq.(52)
$\underline{a}^*$	vector of true parameter values defined in Eq.(87)
$\underline{\underline{A}},\underline{\underline{B}}$	nxn and nxp jacobian matrices of the functions $\underline{f}$ w.r.t. state vectors and parameters resp. defined in Eq. (89 a)
$\underline{\underline{H}}$	pxp nonsingular matrix which is the inverse of $V(\underline{b})$ defined in Eq.(90)

Greek symbols

α, β, γ $\alpha^0, \beta^0, \gamma^0$	unknown parameters and their given estimates involved in the differential equations defined in Eq.(16) resp.
$\underline{\theta}$	m-dimensional vector of parameters defined in Eq.(25)
λ	maximum eigenvalue of the jacobian matrix defined in Eq. (84)
$\underline{\epsilon}$	vector of measurements errors following a multi-normal distribution
ϵ	error introduced in numerical integration.
$\underline{\xi}$	n-dimension perturbation vector caused in the state vector due to changes $\underline{\alpha}$ in $\underline{a}^*$
$\underline{\alpha}$	vector of small changes in parameters in Eq.(88)

ABSTRACT

Mathematical models are used to represent a physical system but for most of the real systems, it is very rare, that a mathematical model represents all the details of the actual process. The models that are based on physical and chemical principles such as conservation of mass, momentum and energy, chemical kinetic mechanisms etc. are, sometimes, called as mechanistic models. Parameter estimation forms an integral part of model building for such cases.

Parameter estimation procedures for algebraic models are quite well established but insignificant literature exists for the case of differential equation models. In the present work, four different algorithms for parameter estimation in differential models are compared for their computational ease and the reliability of the parameter estimates. Approximate analytical solutions are also presented for models linear in state variables.

Two models representing batch reactor kinetic data with eight data sets representing different experimental conditions, were used to test the stability of the algorithms.

Two of the algorithms employed explicit and implicit method of integration respectively, combined with

nonlinear least squares analysis. The third algorithm used the Constrained Simplex method [29] for minimization of an objective function based on variance-covariance matrix of residuals. A weighted residual method was used in the fourth algorithm which gave explicit estimates of parameters, thereby, obviating the necessity of numerical integration and function minimization.

It is observed that the weighted residual method (Algorithm 4) is quite satisfactory for the task of parameter estimation in differential equation models as the parameters obtained from the application of this method had the maximum reliability and minimum computer central processing time. It is hoped that this work would be helpful to the future investigators working with dynamic models.

CHAPTER 1

INTRODUCTION

One of the important aspects of Chemical Engineering systems is their overall design and analysis. These complex systems are represented by mathematical models which may be derived in a variety of ways, for example,

a. They may be specified directly from basic principles of Physics and Chemistry i.e. knowing the underlying physical and chemical principles of the system one can write down the model equations. Models of this type, relevant to chemical engineering systems, are often called " transport phenomena models" (1). Models given by the phenomenological equations of change, that is the continuum equations describing the conservation of mass, momentum, and energy, are examples of transport phenomena models. These models may also be termed as deterministic models as the variables and parameters involved can be assigned a definite fixed number, for any given set of conditions.

b. At the other extreme is the so called " Black Box" model where essentially no a-priori information is available about the system. The model must be determined solely on the basis of the experimental data gathered. Polynomial expressions used to fit the data collected from an experiment represent a class of empirical or " Black Box" models. There is a lot

of uncertainty about these models and extreme caution, therefore, should be employed while using them for the prediction of the state of the system for some other conditions not used at the time of original experimentation.

c. Most real situations lie somewhat intermediate of these two extremes. We often have some information on the basic nature of the system but not enough to set down precisely the relation among all the variables. It is in this case that the modeling procedure often includes extensive comparison of alternate models that can be written based on different underlying mechanisms, to determine the " best model ". This is known as mechanistic model building. It is also in this case that the stochastic models find extensive usefulness. In a typical Chemical Engineering situation, for example, there might exist a mathematical model $\underline{n} = f(\underline{\theta}, \underline{\xi})$ that is perhaps nonlinear in the parameters $\underline{\theta}$, which relates a measurable response n to the controllable variables $\underline{\xi}$. The response might be the rate of a chemical reaction, the variables might be partial pressures of the reactants and products, and the parameters might be adsorption equilibrium constants and an overall reaction rate constant. We may not know the true values of the parameters involved in the model and this introduces some uncertainty in the mechanistic models. In some way this uncertainty is minimized to give a reasonably accurate model for the system

under consideration, which may be used with some degree of confidence for extrapolation.

Once the models are formulated the next important thing is the solution of these models to indicate the nature of the response of the systems concerned. This may require the use of analytical or numerical techniques. Once the response is predicted from a model it may be compared with the experimental data to find out the most plausible model representing the given situation. The techniques involved in the solution of the models would depend upon their types, for example, the relationship between the controllable variables and the dependent variable or response might be represented by algebraic equations-meaning that the response of the system is measured experimentally. In such situations, the comparison between the model prediction and the data can be carried out directly based on some standard statistical techniques such as linear and nonlinear least squares analysis. These models may be of two different types depending upon the complexity of the system studied. For example, if the system considered represents a single reaction between some chemical compounds, carried out in the presence of a catalyst, then the model representing this physical situation would be a " single response model", as the rate measurement of only one component is sufficient to identify the system completely. Take, for example, the catalytic dehydrogenation of ethyl alcohol into

acetaldehyde and hydrogen. Depending on whether 1) the adsorption of ethyl alcohol, A, 2) the surface dehydrogenation, or 3) the desorption of acetaldehyde, R, is the rate determining step, the following mechanistic models of the Hougen-Watson type have been derived [2], in terms of equilibrium constants, rate constants and partial pressures.

$$\underset{A}{C_2H_5OH} \rightleftharpoons \underset{R}{CH_3CHO} + H_2 \tag{1}$$

$$E(r_A) = k(p_A - p_R^2 / K) / (1 + K_A p_R^2 / K + (K_R + K_S) p_R) \tag{2}$$

$$E(r_A) = kK_A(p_A - p_R^2 / K) / (1 + K_A p_A + (K_R + K_S) p_R)^2 \tag{3}$$

$$E(r_A) = kK(p_A - p_R^2 / K) / (p_R + K_A p_A p_R + KK_R p_A + K_S p_R)^2 \tag{4}$$

where $E(r_A)$ = expected value of the rate of ethanol dehydrogenation.

To illustrate the multiresponse situation, we may consider the following moderately complex structure of solid catalyzed chemical reactions [3].

$$\begin{array}{l} A \rightleftharpoons B + C \longrightarrow E \\ A \rightleftharpoons D \end{array} \tag{5}$$

Among many others, three representative rival mechanisms may be thought of:

1) all reactions are described by simple first order homogeneous kinetics i.e., interaction of the species with the catalyst is not explicity accounted for;

2) it is supposed that the reaction $A \rightleftharpoons B+C$ occurs on dual sites, whereas $A \rightleftharpoons D$ and $C \longrightarrow E$ both only require a single site; this model is the true model.

3) all reactions occur on single sites, whereby e.g. the component B is not adsorbed noticeably on the catalyst surface.

The following rate equations are then derived (table 1). The different reaction rates are the responses of the system.

Although there are five components involved, there are only three independent chemical reactions.

Model I	Model II
$r_D = k_1 (p_A - p_D/K)/p_t$	$r_D = k_1 K_A (p_A - p_D/K)/\emptyset$
$r_B = k_2 (p_A - p_B p_C/K')/p_t^2$	$r_B = k_2 K_A (p_A - p_B p_C/K')/\emptyset^2$
$r_C = k_3 p_C/p_t$	$r_C = k_3 K_C p_C/\emptyset$
	$\emptyset = 1 + K_A p_A + K_B p_B + K_C p_C + K_D p_D + K_E p_E$
Model III	**Model IV**
$r_D = k_1 K_A (p_A - p_D/K)/\psi$	$r_D = k_1 K_A (p_A - p_D/K)/\psi$
$r_B = k_2 K_A (p_A - p_B p_C/K')/\psi$	$r_B = k_2 K_A (p_A - p_B p_C/K')/\psi/p_t$
$r_C = k_3 K_C p_C/\psi$	$r_C = k_3 K_C p_C/\psi$

$$\psi = 1 + K_A p_A + K_C p_C + K_D p_D + K_E p_E$$

Table 1. Rival Rate Equations for $A \rightleftharpoons B + C \longrightarrow E$, $A \rightleftharpoons D$

In some situations the equations may be of differential type and involve the controllable variables implicitly. An example representing this situation, is the acid-catalyzed hydrolysis of propylene oxide which, in dilute solution is described by the single reaction [3].

$$CH_3CH\overset{O}{\frown}CH_2 + H_2O \xrightarrow{H+} CH_3CHOHCH_2OH \qquad (6)$$

If the reaction is carried out adiabatically in a batch reactor, the walls of which have negligible heat capacitance, the system equations are of the form:

$$\frac{dy_1}{dt} = -\theta_1 \exp\left(\frac{\theta_2 y_2}{1+y_2}\right) y_1^n \qquad (7)$$

$$\frac{dy_2}{dt} = J\theta_1 \exp\left(\frac{\theta_2 y_2}{1+y_2}\right) y_1^n \qquad (8)$$

with initial conditions

$$\left.\begin{array}{l} y_1 = y_{1o} \\ y_2 = y_{2o} \end{array}\right. ; \quad t = 0 \qquad (9)$$

where y_1 is the molar concentration of propylene oxide, y_2 is a normalised temperature, t is time; θ_1 and θ_2 are undetermined parameters related to the pre-exponential Arrhenius factor k_o and activation energy A by

$$\theta_1 = k_o \exp\left(\frac{-A}{RT_b}\right), \quad \theta_2 = \frac{A}{RT_b} \qquad (10)$$

$J = \dfrac{(-\Delta H)}{\varrho C_p T_b}$ is a thermicity factor, measured independently, and n is the reaction order, assumed to be known.

In the previous example of multiresponse system, the continuity equations for the species D,B and E in a plug flow integral tubular reactor read:

$$\frac{dx_D}{dw/F_{Ao}} = r_D; \quad \frac{dx_B}{dw/F_{Ao}} = r_B; \quad \frac{dx_E}{dw/F_{Ao}} \; r_E; \quad x_D = x_B = x_E = 0 \quad \text{when } w/F_{Ao} = 0 \qquad (11)$$

When the expressions for r_D, r_B and r_E, as given in table 1, are substituted in Eq.(11), we get a set of simultaneous differential equations.

As we see that it is not uncommon to have many rival mechanisms for the same system, due to the uncertainty in the theoretical knowledge of physical situations, it is worthwhile to have a generalised strategy for model discrimination for example in the case of heterogeneous kinetic systems. Such strategy would involve a comparison of several rival kinetic models to select the most " adequate" model, and the estimation of parameters involved in this model. The method might depend upon the way the experimental data are gathered which, in turn, may depend on the way the experiments are planned. There are two ways in which experimentation can be planned so as to allow the maximum discrimination between the rival models - sequential or non sequential experimental designs. The non sequential way would mean the a-priori selection of experimental conditions in terms of controllable variables whereas sequential experimental design strategy

might be used to reduce the experimental effort required at the same time carrying out the discrimination and parameter estimation sequentially. However, this procedure would require a knowledge of a few preliminary data points which may be obtained by using nonsequential fractional factorial designs.

A typical example for the case of model discrimination may arise when an experimentor is studying a chemical reaction in which a reactant A is used to make a product B. Simultaneously, however, an undesired by product C is formed. Suppose further that the experimentor knows that the reaction is one of the following:

$$\begin{aligned} &\text{Mechanism 1 :} \quad A \rightarrow B \rightarrow C \\ &\text{Mechanism 2 :} \quad A \rightarrow B \rightleftharpoons C \end{aligned} \tag{12}$$

If one of these mechanisms is actually correct and all experiments consists of measurements of the concentrations of B at different times, then, where to place the experiments to discriminate most efficiently between the two models will depend in general on the values of the estimated parameters. A sequential scheme therefore suggests itself in which at each stage the two models are refitted and a decision is made as to what experiment should be run next.

Designs for parameter estimation are carried out if the form of the theoretical model is known, the problem

which confronts the experimentor is to evaluate the physical parameters (e.g., the rate constants in chemical kinetics). If the experiments are not carefully planned the experimental points may be so situated in the space of the variables that estimates that are obtained for the parameters are not only imprecise but also highly correlated. Once the data are collected a statistical analysis, no matter how elaborate , can do nothing to remedy this unfortunate situation. However, by selecting a suitable experimental design in advance ,these shortcomings can be overcome. The initial set of experiments could be run according to a standard fractional factorial pattern [4] and then sequential strategy implemented based on the initial estimates obtained.

A flow diagram that illustrates the sequential design procedure for model discrimination and parameter estimation can be drawn as below:

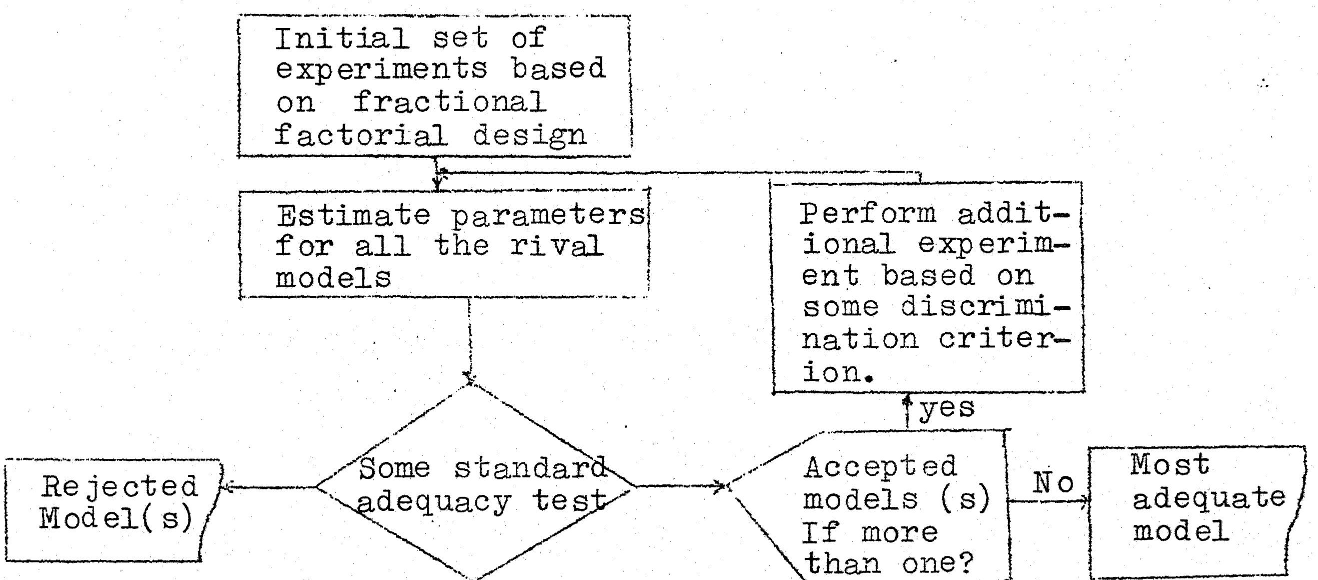


Fig.1 . Sequential Design Strategy for Model Discrimination and Parameter Estimation.

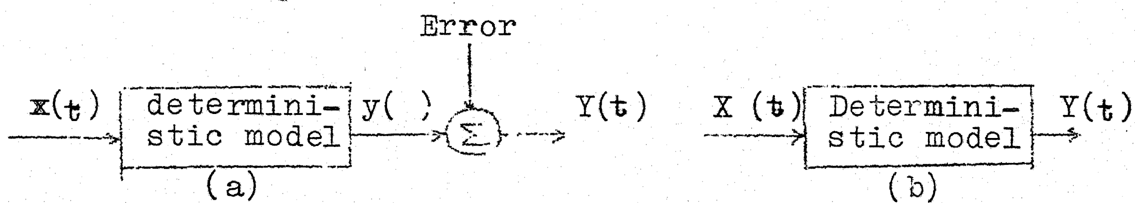
A considerable work has been done in the area of parameter estimation and model discrimination for the case of algebraic models [5-8] but meagre information seems to be available for the case of models governed by differential equations.

The estimation of parameters in ordinary differential equations is of considerable importance, for example, in the determination of kinetic parameters from batch and integral reactor data, studying the dynamics of systems such as missiles and spacecraft trajectories from tracking data, and in several other occasions where the rate of change of state variables w.r.t. some independent variable cannot be measured directly. The procedure, in many cases, is usually the extension of the principles used for the case of algebraic equations. However , the task of parameter estimation and model discrimination, in the case of the former which is quite complicated due to the multiresponse systems involved, is further complicated due to the nonavailability of the responses directly from model thus making it necessary either to integrate the differential equations or to use some variational principles which may circumvent the integration of the equations. A redefinition of dependent and independent variable would be appropriate at this stage. The dependent variable for the case of algebric equations is replaced by the derivative of the state variable in case of differential models; for example, it could be the

immeasurable rate of change of the concentration of reactants and products w.r.t. time or space which, incidently, are the independent variables in differential equations. The concentrations or such variables in which a change is observed w.r.t. independent variables, are known as state variables as they determine the state of the system at a given time or space. The experimental data that is collected, in this situation, may be a continuous function of time and / or space or might have been gathered only at discrete points. The statistical analysis of these two types of situations is based on concept of " noise" that gets added to the data during measurements due to the inaccuracy of measuring instruments and human oversight. This "noise" may be of two types as illustrated below:

a. Noise in the output of the system; for example, random errors in the measurements of reaction rate or concentrations in a reaction kinetic study.

b. Dynamical noise that gets added to the input of the system implying that the differential equations act on this noise too, along with the state variables. A pictorial representation would be:



where the bold letters denote random variables and stands for the error structure superimposed on the output of the system.

We will deal only with such systems that are represented by ordinary differential equations and do not involve any dynamical noise, as these systems are quite common in kinetic reaction studies of complex reaction networks.

LITERATURE REVIEW:

The advent of high speed digital computers has seen a large number of research papers published in the area of parameter estimation and model discrimininination for the case of algebraic models but the literature on parameter estimation and model discrimination in multiresponse systems and differential equation models appears to be meagre.

The problem of parameter estimation can be posed as follows:

The system is described by a vector of ordinary differential equations

$$\underline{\dot{X}} = \underline{f}\,(\underline{X},\underline{k}\,) \qquad \underline{X}(0) = \underline{X}_0 \tag{13}$$

where $\underline{X}$ is an n-dimensional state vector and $\underline{k}$ is the p-dimensional parameter vector. The output of the system is the m-dimensional vector $\underline{y}$. Measurements of the output are

made at R times $t_1, t_2, \ldots, t_R$. The measurements are related to the state vector according to

$$\underline{Y}_r = \underline{h}\ (t_r, \underline{X}\ (t_r) + (\text{ Measurement Errors}) \quad (14)$$

$$r = 1,2,\ldots, R.$$

Given the observations $\underline{Y}_r$ we want to fit the model given by Eq.(13) by the data $\underline{Y}_r$ $r = 1,2,..,R$ in some optimal way.

If we assume the initial state of process $\underline{X}_o$ to be known, we desire to select the parameters $\underline{k}$ to minimize the least squares criterion (if $\underline{X}$ is also unknown we will want to select both $\underline{X}_o$ and $\underline{k}$)

$$S= \sum_{r=1}^{R} \| \underline{Y}_r - \underline{h}\ (t_r, \underline{X}\ (t_r)) \|^2_{\underline{\underline{Q}}_r} \quad (15)$$

where $\underline{\underline{Q}}_r$ is an mxm positive semidefinite weighting matrix incorporating the relative accuracy of each component of the measurement vector at each time of measurement.

There are essentially eight different ways by which we can estimate the parameters involved in heterogeneous reacting systems for differential types of models, from experimental data. These are discussed in the literature as follows:

1. Analytical (exact or approximate) integration of the set of differential equations, and subsequent application of iterative nonlinear least squares regression techniques[5-9]. Kittrell et.al [5] have proposed a method, which is a variation of the method of integration, that transforms the dependent

variable to achieve an error distribution as consistent as possible with the assumptions inherent in the least squares analysis. The method can be applied to the determination of unknown reaction order and rate constants for power law rate models very successfully and for a small set of equations.

2a. Differentiation of the emprical data directly, and subsequent application of linear least squares regression techniques (10,7,11,12). Here one has to bear in mind that as differentiation is more error prone than the original data and so the estimates obtained, may not be the optimal estimates of the rate constants. Thus, differentiation of empirical data increases the existing inherent errors in the experimental data and therefore, category 2 is not satisfactory.

2b. Another procedure that can be followed based on differentiation is to use linear regression to fit the emprical data, differentiation of the regression equation followed by linear regression to estimate the coefficient [13].

3a. Numerical integration of the set of differential equations using emprical data directly, followed by iterative non-linear least squares technique [14-17]. Ball and Groenweghe[14] have used the values of the constants, obtained by the differentiation of the data and subsequent use of linear least squares equation, as the initial guess values for the iterative numerical integration procedure.

3b. Linear regression to fit the emprical data, followed by numerical integration of the set of differential equations.

4. Trial and error search using analog computers to match the emprical data [18,19].

5. Method of differential corrections may also be used [20]. Incase, if the differential equations admit explicit solutions in terms of known elementary or transcendental functions then according to the method, if

$$y = f\,(\,t, \alpha, \beta, \gamma)$$

is the solution of the differential equation and if $\alpha^{(o)}$, $\beta^{(o)}$ and $\gamma^{(o)}$ are given estimates of the unknown parameter values, the n equations

$$f\,(t, \alpha, \beta, \gamma\,) = f\,(t, \alpha^{(o)}, \beta^{(o)}, \alpha^{(o)}) + \sum_{k=1}^{m} \frac{\partial f}{\partial \alpha_k} (\alpha_k - \alpha_k^{(o)}) + \sum_{k=1}^{m} \frac{\partial f}{\partial \beta_k} (\beta_k - \beta_k^{(o)}) + \sum_{k=1}^{m} \frac{\partial f}{\partial \gamma_k} (\gamma_k - \gamma_k^{(o)}) \tag{16}$$

are formed where the derivatives are evaluated at $(t, \alpha^{(o)}, \beta^{(o)}, \gamma^{(o)})$. The equations give the corrections to the first approximation and then, the new values of the estimates are used and the process continued iteratively. The method may be employed in the case of differential being nonlinear for which **explicit** solutions are not available.

6. Quasilinearisation [21,22] is another technique which has been used to tackle the problem. In a situation where we have n dimensional state vector and p dimensional parameter vector, we define

$$\underline{z} = \binom{\underline{x}}{\underline{k}} , \quad \underline{g} = \binom{\underline{f}}{o} \tag{17}$$

$$\text{Since } \underline{k} \text{ is constant, } \frac{d\underline{k}}{dt} = 0 \tag{18}$$

We adjoin Eq. (18) to Eq. (13) and use Eq. (17) to give

$$\dot{\underline{z}} = \underline{g}\ (t,\underline{z}) \quad ; \quad z_i(o) = x_{io} , \quad i = 1,2,.., n ;$$

$$z_i(o) = \ ? \quad , \quad i = n+1,\ n+2,.., n+p$$

The problem is to find the initial conditions $z_i(o)$, $i = n+1,..n+p$ to minimize the least squares criterion

$$S = \sum_{r=1}^{R} (\underline{y}_r - \underline{h}\ (t_r,\underline{z}\ (t_r;z(o))))^T Q_r\ (\underline{y}_r-\underline{h}\ (t_r,\underline{z}\ (t_r;z(o))))$$

where $\underline{z}\ (t;z(o))$ is the solution of Eq. (13) corresponding to a particular set of $z_i(o)$, $i = n+1,.., n+p$.

7. Certain short cut methods have also been presented in the literature [23,24]. These methods are very useful for parameter estimation as the estimates are obtained without solving the differential equations. In [23], the mathematical model is assumed to be linear in parameters and the model is composed of M independent equations of the form:

$$\frac{dC_1}{dt} = B_1R_{1,1}(t) + .. + B_kR_{1,k}(t) + .. B_NR_{1,N}(t); C_1(o)=C_{o1}$$

.

.

$$\frac{dC_j}{dt} = B_1R_{j,1}(t) + .. +B_kR_{j,k}(t) +.. B_NR_{J,N}(t); C_J(o)=C_{o2} \quad (19)$$

.

.

$$\frac{dC_M}{dt} = B_1R_{M,1}(t) + .. +B_kR_{M,k}(t) +.. B_NR_{M,N}(t); C_M(o)=C_{oM}$$

where the C's (concentrations) are the dependent variables, the B's are the kinetic coefficients, presumed constant, and the R's represent any desired function of C and t (many of which may be zero). The Eq. (19) can be written in an abbreviated form as $\frac{dC_J}{dt} = \sum_{k=1}^{N} B_kR_{J,k}$ (J= 1,2, .., M) (19a)

Eq. (19) has a unique solution if the right hand side of Eq.(19a) has a continuous uniformly bounded partial derivatives with respect to C_j in the region of interest. Now the main difficulty in using Eq.(19) directly to find the B_k (problem in the category 2 above) is that, as discussed above, the derivatives on the left hand side must be known and this cannot be done accurately. On the other hand the formal integration requires iterative calculations (category 3). However, if each equation is directly integrated using the fact that the right hand side involves only integration of functions of the measured concentrations as functions of time, good features of both categories 2 and 3 are retained.

Thus, after integration from t_o to t_i Eq. (19a) may be written as

$$C_j(t_i) - C_j(t_o) = \sum_{k=1}^{N} B_k \int_{t_o}^{t_i} R_{j,k}\, dt \qquad \begin{matrix}(i= 1,2,..,p)\\(j= 1,2,..,M)\end{matrix} \qquad (20)$$

where P is the number of time intervals between the times at which the concentrations were measured. Eq.(20) may be abbreviated as $Y_{ij} = C_{ij}-C_{1j} = \sum_{k=1}^{N} B_k X_{ijk}$ (20a)

The r.h.s. gives the predicted value of the modified dependent variable and observed values can be obtained from the experimental data. A least square criterion may be then used to estimate the unknown coefficients, B_k. It is better to use sum of squares of the weighted deviations, $DN_{ij} = (Y_{ij}-\hat{Y}_{ij})W_{ij}$, instead of sum of squares of the deviations $(Y_{ij}-\hat{Y}_{ij})$, where W_{ij} is the weight associated with each deviation.

8. Sequential Experimental Design procedures for precise parameter estimation in ordinary differential equations were first presented by Hosten and Emig [37]. These designs are based on the ellipsoid region that surrounds the true values of the parameters. Box and Lucas[38] suggested to select those experimental conditions which minimize the volume of the joint confidence region. This is realised when det $V(b)^{-1}$ or equivalently, det (X^TX) is maximized. This criterion is known as " Minimum Volume" criterion. V(b) is the variance-covariance matrix of the parameters.

Hosten [39] advocated realising the greatest possible contraction of the longest principal axis of the hyperellipsoid, rather than trying to attain the smallest possible dimen ion. It is done by performing those experiments which maximize the smallest eigenvalue of $V(b)^{-1}$. This criterion which acts upon the shape of the joint confidence region is known as " shape " criterion. These designs determine the influence of the sampling range on the precision of the parameters and also that of the number of sampling points in each range.

CHAPTER 2

PARAMETER ESTIMATION IN ORDINARY DIFFERENTIAL EQUATIONS

The problem of parameter estimation in case of algebraic models has received considerable attention in the past and the results are quite well established on the basis of sound statistical principles.

For a linear algebraic model the following basic formulas are well known [25]

$$\underline{Y} = \underline{X}\,\underline{\beta} + \underline{\epsilon} \tag{21}$$

$$\underline{b} = (\underline{X}^T\underline{X})^{-1}\,\underline{X}^T\underline{Y} \tag{22}$$

$$V(\underline{b}) = (\underline{X}^T\underline{X})^{-1}\,\sigma^2 \tag{23}$$

$$(\underline{\beta}-\underline{b})^T\,\underline{X}^T\underline{X}\,(\underline{\beta}-\underline{b}) = s^2 pF\,(\,p, n-p, 1-\alpha\,) \tag{24}$$

Eq. (21) relates the n observations $\underline{Y}$ of the dependent variable $\underline{Y}$ to the nxp values of the independent variables $\underline{X}$ Eq. (22) gives the least squares estimates $\underline{b}$ of the unknown parameters

Eq. (23) and (24) are respectively the expressions for the covariance matrix of the estimates $\underline{b}$ and the expression for the joint confidence region E associated with $\underline{b}$. Eq.(23) and (24) are based on the assumption that the experimental errors are independent with zero mean and constant variance σ^2. In addition Eq. (24) requires that $\underline{\epsilon}$ is multinormally distributed i.e., $\underline{\epsilon}$ $N(\underline{0},\underline{I}\,\sigma^2)$. s^2 is the estimate of the error variance σ^2, obtained from the model.

An important problem in process analysis is the estimation of parameters in a differential model from experimental data. Often it is impossible to solve the differential equations explicitly and one is forced to estimate parameters directly from the differential equations.

One way of going about this is to combine numerical integration of the differential equations with non linear least squares estimation of parameters, for which there are several algorithms available.

Alternatively, the differential equations may be linearised and the parameters determined by a repeated application of linear least squares, an approach which forms the basis of the quasilinearisation technique as explained earlier.

Variational principles may prove very helpful in circumventing the cumbersome job of repeatedly integrating a set of differential equations and function minimisation and save a lot of computer time. Indeed, in many cases, when the equations are linear in parameters, these methods provide <u>explicit</u> estimates of parameters. Weighted residual methods fall in the category of these methods.

Looking at the importance of parameter estimation in ordinary differential equations it was decided to use different algorithms that are available for parameter

estimation in ordinary differential equations, and compare their suitability for a particular problem at hand.

Four different algorithms are compared for their fastness and reliability of the parameters estimated by them. The models, used for testing the efficiency of the algorithms represent batch reactor kinetic data for some of the most important industrial reactions viz. direct esterification of terephthalic acid by ethylene oxide to produce bis-hydroxy terephthalate and isomerisation of m-xylene [26,27]. In case of batch reactor, the models representing the kinetics of the reactions are given by first order ordinary differential equations and the problem is essentially an initial value problem. The choice of the algorithms was based on the different theoretical considerations that exist for solving differential equations and their ease of applicability.

Runge-Kutta methods have an important place in the numerical integration of differential equations especially for initial value problems. An algorithm which uses a fourth order explicit Runge-Kutta-Gill algorithm [36] was used to estimate the parameters of the models for two different kinetic schemes. These methods are quite sensitive to the stepsize of integration and the stiffness of the differential system. These problems are discussed, in detail, later. Because of its very frequent use nonlinear least squares analysis, was joined to the numerical integration scheme for the minimisation of

the sum of squares of the deviations of the predicted values of concentrations from the measured values.

Another minimisation program based on the Box-complex method [29] was also tried to determine the effect of the efficiency minimisation on the estimated values of the rate constants. The algorithm was used in conjunction with a fourth order explicit integration scheme.

Orthogonal collocation methods [32] are used quite frequently for the numerical solution of differential equations. The applicability of the technique to initial value problems-especially nonlinear first order differential equations leads to one point collocation method which is quite suitable for nonlinear coupled systems as the method is an implicit one. An algorithm based on an implicit second order scheme for numerical integration, coupled with nonlinear least squares analysis was used to observe an improvement in the values of the parameters by using implicit equations. These methods are more stable and reliable at the expense of more computational time.

The fourth algorithm used was based on variational principles and used weighted residual method for parameter estimation directly without solving the set of differential equations. The method provides with an explicit estimates of parameters for models linear in parameters and even for nonlinear models obviates the necessity of minimisation of an

objective function and numerical integration. A set of non-linear algebraic equations result which can be solved by using Newton-Raphson or successive substitution techniques. The method is extraordinarily fast and requires a negligible computer time.

Let us define the problem first before suggesting the different ways to solve it.

Consider the problem of estimating the vector of parameters $\underline{\theta}$ in the differential model

$$\frac{dy}{dt} = \underline{f}\,(\underline{y},\underline{\theta}) \tag{25}$$

with initial conditions $\underline{y} = \underline{y}_0$, $t = 0$

where

(i) $\underline{y}$ is the n-dimensional state vector

(ii) $\underline{\theta}$ is the m-dimensional vector of parameters

(iii) $\underline{f}$ is the n-dimensional vector function of <u>known</u> model forms.

(iv) t is the independent variable .

As can be seen the equations are most general in nature and can represent continuous flow integral reactors with t representing reactor length, Z.

Suppose we are given a set of observations of $\underline{Y}(\underline{y})$ at p discrete values of t, assumed related to the state by

$$\underline{Y}_r = \underline{y}_r + \underline{\epsilon}_r \quad (r= 1,2,\ldots ,p) \tag{26}$$

where $\underline{\epsilon}_r$ is an n-dimensional vector of observation errors.

ALGORITHM 1:

Fourth order explicit Runge-Kutta-Gill method of integration (36) is used to solve the given set of differential equations and then the value of the state vector $\underline{Y}$ is used in Eq. (26) to calculate the vector of residuals. To obtain the estimates of parameters, the objective function given by resideual sum of squares is minimized by using a nonlinear least square minimisation routine.

The R-K-G method used is described by

$$\frac{d\underline{y}}{dt} = \underline{f}\ (\underline{y}, \underline{\theta})$$

assuming that $\underline{\theta}$, the vector of parameters is known

$$\underline{y}_{n+1} = \underline{y}_n + \frac{h}{6} (K_1+2K_2+2K_3+K_4)$$

where
$$K_1 = \underline{f}\ (\underline{y}_n) \tag{27}$$

$$K_2 = \underline{f}\ (\underline{y}_n+ \tfrac{1}{2}\ h\ K_1)$$

$$K_3 = \underline{f}\ (\underline{y}_n+AK_1+BK_2)$$

$$K_4 = \underline{f}\ (\underline{y}_n+CK_2+DK_3)$$

where A,B,C and D are R-K-G constants given by

$$A=\frac{\sqrt{2}-1}{2} \qquad B= \frac{2-\sqrt{2}}{2} \qquad C = \frac{-\sqrt{2}}{2} \qquad D= 1+ \frac{\sqrt{2}}{2}$$

Once the vector of state variables at different time values

is known, residual sum of squares is evaluated which is a measure of discrepancy between the observed values and the model values is calcuated

$$\underline{Y}_r = \underline{y}_r + \underline{\epsilon}_r \qquad (r = 1,2,..p)$$

$$\text{R.S.S.} \quad = \sum_{s=1}^{n} \sum_{r=1}^{p} (Y_{rs} - y_{rs})^2 \tag{28}$$

This objective function is minimised by using a nonlinear least squares minimization routine.

In the flow diagram that illustrates the algorithm:

y_{rs} - observations for the multiresponse system with $r = 1,..,p$ and $s = 1,..,n$

f_s - vector of functions of $\underline{y}$ and $\underline{\theta}$

$\underline{\theta}_o$ - vector of initial guesses of parameters

y_{so} - initial conditions for the n-responses

t_r - discrete time values at which measurements are done

p - number of observations.

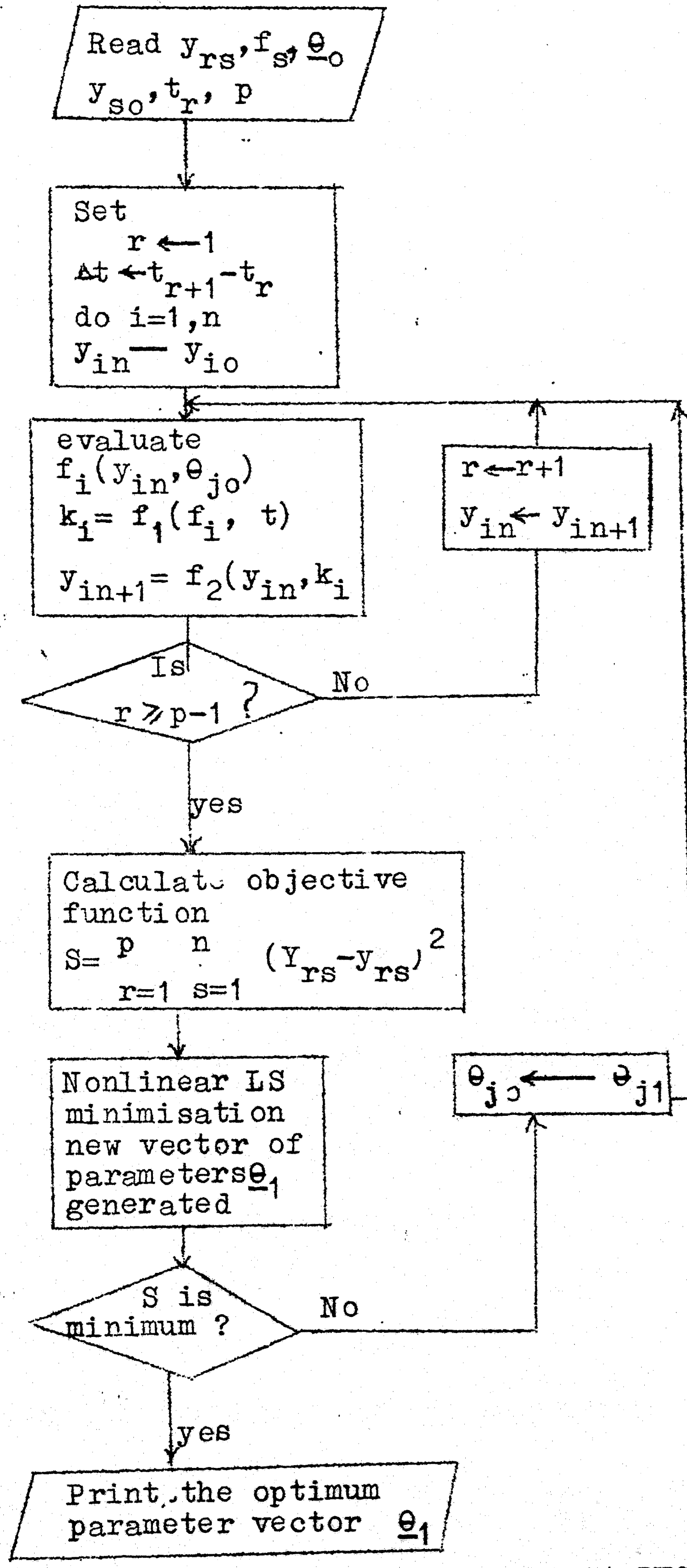


Fig. 2 Parameter estimation in ODEs by explicit RKG method.

Non Linear Least Squares Analysis:

A modification of Gauss- Newton method as suggested by Gill and Murray is used in the present work for the solution of nonlinear least squares problem [28]. The algorithm is concerned with the location of a point $\underline{x}^*$ which minimizes the sum of squares of nonlinear functions

$$F(x) = \sum_{i=1}^{m} (f_i(\underline{x}))^2, \quad \underline{x} \in E^n, \; m \geq n \qquad (29)$$

The gradient vector $g(x)$ and Hessian matrix $\underline{\underline{G}}(x)$ of $F(x)$ are given by $2J(x)^T f(x)$ and $2(J(x)^T J(x) + B(x))$ respectively where $J(x)$ is the mxn jacobian matrix of $f(x)$ whose ith row is $\nabla f_i(x) = (\partial f_i/\partial x_1, \partial f_i/\partial x_2, \ldots, \partial f_i/\partial x_n)$, $B(x)$

$= \sum_{i=1}^{m} f_i(x)\, G_i(x)$ and $G_i(x)$ is the Hessian matrix of $f_i(x)$ ($F(x)$ assumed to be twice-continuously differentiable). For unconstrained minimisation problems where the Hessian matrix of second derivatives can be used, Newton's Method can be used. The method constructs a sequence of vectors

$$\{x^{(k)}\} \text{ such that } \quad x^{(k+1)} = x^{(k)} + \alpha^{(k)} P_N^{(k)} \qquad (30)$$

where $\alpha^{(k)}$ is a scalar steplength and $P_N^{(k)}$, the direction of search, satisfies the equation

$$G(x^{(k)})\, P_N^{(k)} = -g(x^{(k)}) \qquad (31)$$

The special form of the Hessian matrix and gradient vector can be used in the Newton equation to give the equivalent

form $(J(x^{(k)})^T J(x^{(k)}) + L(x^{(k)}))\ P_N^{(k)} = -J\ (x^{(k)})^T f(x^{(k)})$ (32)

The Gauss-Newton Method exploits the special structure of the Hessian matrix and gradient vector which occurs with least squares problem. The method computes the direction of search as the solution of

$$J(x^{(k)})^T J(x^{(k)})\ p_{GN}^{(k)} = -J\ (x^{(k)})^T\ f\ (x^{(k)}) \tag{33}$$

These equations are obtained by neglecting the second derivative matrix $B(x^{(k)})$ in Eq.(32). The Gauss-Newton method is intended for problems where $\| B(x) \|$ is small compared to $\| J(x)^T\ J(x) \|$ such as the so-called " small residual problem" where $f(x) \rightarrow o$ as $x \rightarrow x^*$.

Eq. (33) are the so-called " Normal Equations" for the linear least-squares problem.

$$\underset{p}{\text{minimize}} \quad \| J(x^{(k)})\ p+f(x^{(k)}) \|_2 \tag{34}$$

that value of p is selected which minimizes the linear least squares problem and has least Euclidean length . Further insight into the method can be obtained by referring to [28] .

ALGORITHM 2:

The method uses a fourth order Runge-Kutta-Gill scheme for numerical integration of differential equations coupled with the complex method (constrained simplex method) proposed by Box [29]. The complex method searches

for the maximum value of a function $f(x_1, \ldots, x_n)$ subject to m constraints of the form $g_k \leqslant x_k \leqslant h_k$, $k = 1, \ldots, m$, where $x_{n+1}, \ldots, x_m$ are functions of $x_1, \ldots, x_n$, and the lower and upper constraints g_k and h_k are either constants or functions of $x_1, \ldots, x_n$. It has been developed from the simplex method[30]. It requires an initial point $x_1^o, \ldots, x_n^o$ which satisfies all the m constraints to start the search for the maximum (minimum with-f) of a nonlinear function. In this method $k \geqslant n+1$ points are used, of which one is the given initial point. The further k-1 points required to set up the initial configuration are obtained one at a time by the use of pseudo-random numbers and the ranges for each of the independent variables, viz., $x_i = g_i + r_i(h_i - g_i)$ where r_i is a pseudo-random deviate rectangularly distributed over the interval (0,1). A point so selected must satisfy the explicit constraints, but need not satisfy all the implicit constraints. If an implicit constraint is violated the trial point is moved halfway towards the centroid of those points already selected (where the given initial point, is included). Ultimately a satisfactory point will be found. Proceeding this way (k-1) points are found that satisfy all the constraints. The method uses α, the reflection factor as 1.3 and k, the number of vertices for contrained problems, as 2n.

ALGORITHM 3:

This algorithm is well suited for the solution of general first order initial value problems and can handle stiff systems as well. A brief description of the method will give a deeper insight in to the use of the method.

Let us first consider scalar differential equation

$$\frac{dy}{dx} = f(y) \qquad y(x_n) = y_n$$

where f(y) does not depend explicitly on x. A Taylor series from (x_n, y_n) is

$$y_{n+1} = y_n + hf + \frac{1}{2} h^2 f_y f + \frac{1}{6} h^3 (f_{yy} f^2 + f_y^2 f + \ldots) \tag{35}$$

where $f = f(y_n)$, $f_y = \left(\frac{\partial f}{\partial y}\right)_{y_n}$ etc.

Now a one point Gauss method with collocation at $x = x_n + h/2$ and extrapolation to y_{n+1} gives

$$2(y_{1/2} - y_n) = h\, f(y_{1/2}) \tag{36}$$

$$y_{n+1} = y(x_n + h) = y_n + 2(y_{1/2} - y_n) \tag{37}$$

The method is $O(h^3)$ in y_{n+1} but one nonlinear equation $2(y_{1/2} - y_n) = h\, f(y_{1/2})$ should be solved for $y_{1/2}$. If f (y) is linearised from y_n, we obtain

$$2(y_{1/2} - y_n) = h\,(f(y_n) + f_y\,(y_{1/2} - y_n)) \tag{38}$$

or

$$y_{1/2} - y_n = (2 - h\, f_y)^{-1}\, h\, f \tag{39}$$

and finally

$$y_{n+1} = y_n + (1 - \frac{1}{2} f_y h)^{-1} hf \tag{40}$$

Expanding the above equation yields.

$$y_{n+1} = y_n + h f + \frac{1}{2} h^2 f_y f + \frac{1}{4} h^3 f_y^2 f \tag{41}$$

Continuing next to M coupled differential equations we obtain an analogous expression

$$\underline{K}_1 = (\underline{\underline{I}} - 0.5 h \underline{f}_y)^{-1} h \underline{f} (\underline{y}) \tag{42}$$

$$\underline{y}_{n+1} = \underline{y}_n + \underline{K}_1 \tag{43}$$

It is a semi-implicit Runge-Kutta method and works very well even for stiff equations and represents one of the best methods to solve an initial value problem. To solve an extraordinary stiff differential equation an extension of the above method may be used which will have more favorable damping properties:

$$K_1 = (1 - ahf_y)^{-1} hf \tag{44}$$

$$K_2 = (1 - ahf_y)^{-1} K_1 \tag{45}$$

$$y_{n+1} = y_n + R_1 K_1 + R_2 K_2 \tag{46}$$

where the constants a, R_1 and R_2 have the values of

$a = 1 - \sqrt{1/2}$ $R_1 = 1 - \sqrt{1/2}$, $R_2 = \frac{\sqrt{2}}{2}$

<u>ALGORITHM 4</u> :

<u>Weighted Residual Methods:</u>

A vector of equation residuals $\underline{R}$ is defined for the given set of differential equations by

$$\underline{R} = \frac{d\underline{y}}{dt} - \underline{f}\,(\underline{y},\underline{\theta}) \tag{47}$$

the solution of the set of differential equations satisfies the condition $\underline{R} = 0$ identically for all $t \geqslant 0$. It must also satisfy the condition

$$\int_{t_a}^{t_b} \underline{\underline{G}}\,\underline{R}\,dt = 0 \tag{48}$$

where $\underline{\underline{G}}$ is an nxn weighting matrix and t_a and t_b are any t 0. The various weighted residual methods differ only in their choice of $\underline{\underline{G}}$.

The Residual Least Squares Method (RLSM):

Instead of Eq. (48) we may use

$$\min_{\underline{\theta}} \left\{ S = \int_{t_a}^{t_b} \underline{R}^T\,\underline{\underline{W}}\,\underline{R}\,dt \right\} \tag{49}$$

where $\underline{\underline{W}}$ is an nxn weighting matrix. Minimisation of S leads to m normal equations:

$$\frac{\partial S}{\partial \theta_i} = 0 \quad (i = 1, 2, \ldots, m) \tag{50}$$

Suppose we choose $\underline{\underline{W}} = \underline{\underline{I}}$, then Eq. (50) reduces to Eq. (48) where

$$\underline{\underline{G}} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_1} & \cdots & \frac{\partial f_n}{\partial \theta_1} \\ \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_2}{\partial \theta_2} & \cdots & \frac{\partial f_n}{\partial \theta_2} \\ \vdots & & & \\ \frac{\partial f_1}{\partial \theta_m} & \frac{\partial f_2}{\partial \theta_m} & \cdots & \frac{\partial f_n}{\partial \theta_m} \end{bmatrix}$$

combining Eqs. (47) & (48) we obtain

$$\int_{y_a}^{y_b} \underline{\underline{G}} \; d\underline{y} = \int_{t_a}^{t_b} \underline{\underline{G}} \; \underline{f} \left(\underline{y}, \underline{\theta} \right) dt \tag{51}$$

substituting observations $\underline{Y}$ for the state $\underline{y}$ Eq. (51) reduces to a system of nonlinear algebraic equations in the θ_i

$$\sum_{r=o}^{p-1} \bar{\underline{\underline{G}}}_r \Delta \underline{Y}_r = \sum_{r=o}^{p-1} \bar{\underline{\underline{G}}}_r \; \underline{f} \left(\underline{Y}_r, \hat{\underline{\theta}} \right) \Delta t_r \tag{52}$$

However, if the function $\underline{f}$ is given by $\underline{f} = \underline{\underline{A}} \; (\underline{y}) \cdot \underline{\theta}$ then Eq. (52) becomes explicit in $\underline{\theta}$:

$$\underline{\theta} = \left(\sum_{r=o}^{p-1} \left(\underline{\underline{A}}^T \underline{\underline{A}} \right) \; \Delta t_r \right)^{-1} \cdot \left(\sum_{r=o}^{p-1} \underline{\underline{A}}^T \Delta \underline{Y}_r \right) \tag{53}$$

<u>Approximate Analytical Solutions:</u>

The differential equations of first order can be solved analytically but for a set of ordinary differential equations that are coupled, only approximate analytical solutions are possible. These solutions can be utilised very well to prepare a general purpose program for parameter estimation in the models governed by linear differential equations.

A model governed by first order linear differential equations is given by:

$$\frac{d\underline{Y}}{dt} = \underline{\underline{A}} \; \underline{Y} \quad ; \quad \underline{Y} \; (0) = \underline{Y}_o \; (\text{ Given})$$

This vector equation can be solved analytically to give

$$\underline{Y}(t) = e^{\underline{\underline{A}}(t-t_o)}\,\underline{Y}_o = \exp(\underline{\underline{A}}(t-t_o))\,\underline{Y}_o \tag{54}$$

The exponential matrix is defined by the infinite series

$$e^{\underline{\underline{A}}t} = \underline{\underline{I}} + \underline{\underline{A}}t + \underline{\underline{A}}^2\frac{t^2}{2!} + \underline{\underline{A}}^3\frac{t^3}{3!} + \ldots \tag{55}$$

Let us define the state transition matrix as

$$\underline{\underline{\emptyset}}(t-t_o) = e^{\underline{\underline{A}}(t-t_o)} \tag{56}$$

Therefore, we have

$$\underline{Y}(t) = \underline{\underline{\emptyset}}(t-t_o)\,\underline{Y}_o \tag{57}$$

where we may use a truncated series expansion of the exponential matrix as an approximation to $\underline{\underline{\emptyset}}$ matrix. This matrix is made up of terms containing parameters and thus is a " constant" matrix. For different values of t, one gets the state of the system which undergoes transition .

An algorithm would make the implementation of the procedure more clear (Fig.3).

The responses were calculated both based on $(t-t_o)$ where t is constantly increasing and also based on $(t_{i+1}-t_i)$ where t_{i+1} and t_i are consecutive two time values. The second procedure gave better estimates as was evident from the sum of squares at the end of the program.

The values of the parameters obtained by using this analytical procedure may be used with better degree of confidence than those obtained by using purely numerical

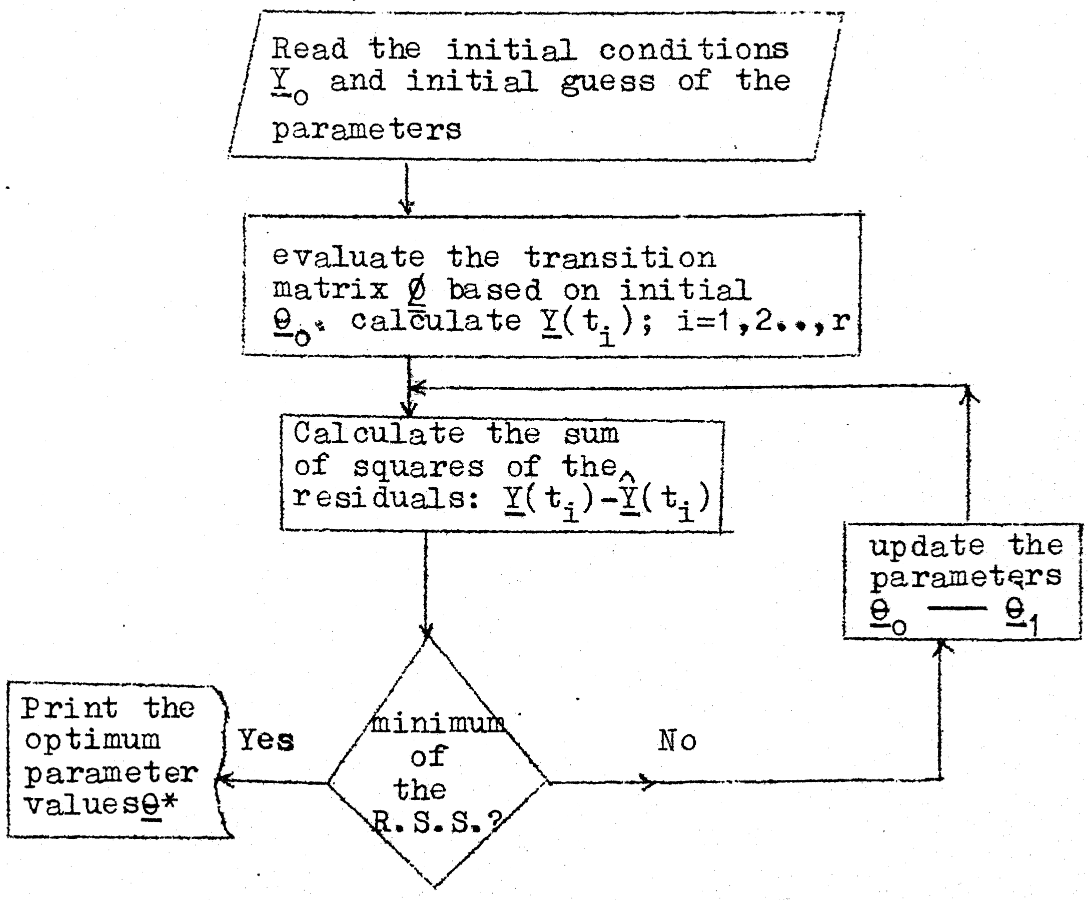


Fig. 3. Algorithm for the implementation of analytical solutions.

techniques. A still better approximation to the optimum may be obtained by using a different objective function which would take into account the covariance of the residuals, especially in the case of multiresponse models.

Formulation of the Objective Function:

Another objective function based on the discrimination criterion of Box [31] was constructed and used in parameter estimation. The results were astoundingly better as was evident from the comparison of residual sum of squares with that of this objective function.

From the theory of linear regression analysis, it is known that the determinant of the matrix $\underline{\underline{X}}^T\underline{\underline{X}}$, where $\underline{\underline{X}}$ is the design matrix of independent variables as pointed out earlier, can be used as a measure of the reciprocal of the variance-covariance matrix of the parameters in case of models governed by $\underline{Y} = \underline{\underline{X}}\,\underline{B}$. Thus, the determinant of $(\underline{\underline{X}}^T\underline{\underline{X}})^{-1}$ may be used as a measure of the deviation of parameters from their true values and thus, may serve as a very relevant objective function. The objective function used is given by:

$$S = \begin{vmatrix} \sum_{i}^{N}(y_{i1}-\hat{y}_{i1})^2 & \sum_{i=1}^{N}(y_{i1}-\hat{y}_{i1})(y_{i2}-\hat{y}_{i2}) & \cdots & \sum_{i=1}^{N}(y_{i1}-\hat{y}_{i1})(y_{iM}-\hat{y}_{iM}) \\ \sum_{i=1}^{N}(y_{i2}-\hat{y}_{i2})(y_{i1}-\hat{y}_{i1}) & \sum_{i=1}^{N}(y_{i2}-\hat{y}_{i2})^2 & \cdots & \sum_{i=1}^{N}(y_{i2}-\hat{y}_{i2})(y_{iM}-\hat{y}_{iM}) \\ \vdots & & & \\ \sum_{i=1}^{N}(y_{iM}-\hat{y}_{iM})(y_{i1}-\hat{y}_{i1}) & \cdots & \cdots & \sum_{i=1}^{N}(y_{iM}-\hat{y}_{iM})^2 \end{vmatrix} \tag{58}$$

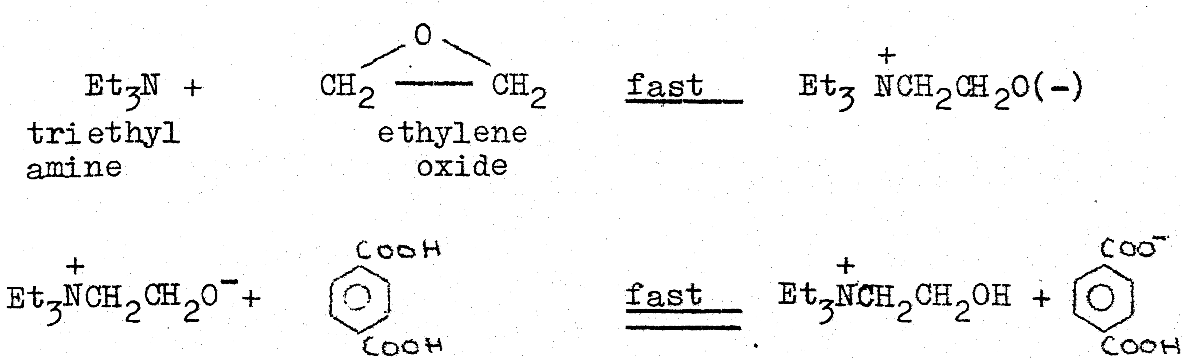
The algorithms discussed above along with the objective function constructed above were used on the real time data available from the literature (26,27) so that the results obtained may be of some practical significance. The data used is reported.

EXAMPLE 1 :

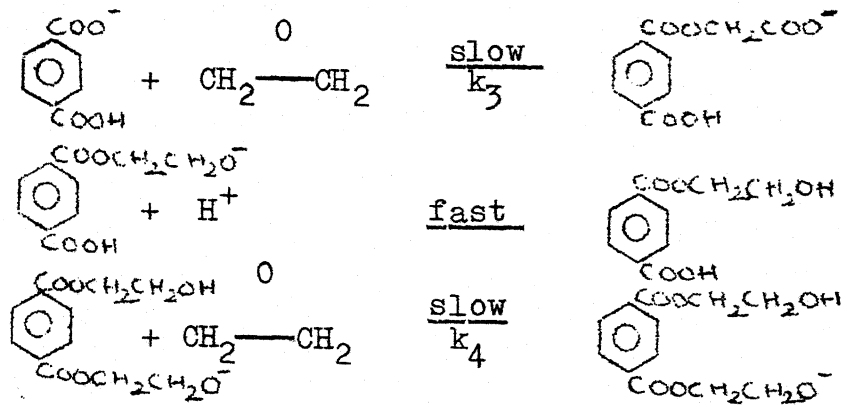
Direct Esterification of Terephthalic acid with Ethylene Oxide:

This reaction is homogeneously catalysed by triethyl amine (atertiary amine) and the reaction mechanism is given below.

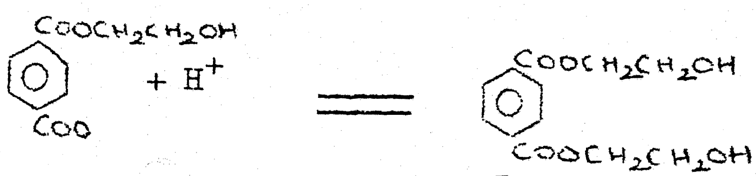
The reaction between ethylene oxide and tertiary amine gives rise to the formation of intermediate (quarternary base salt) which is immediately neutralised by Terephthalic acid.



Terephthalate anion thus formed reacts with ethylene oxide to give 2 hydroxyethyl terephthalate.



Hydroxy ethyl
terephthalate
ion



BHET

Further, the equilibrium constant of the salt formation from a weak acid and a weak base is expressed as ;

$$RCOOH + R'NH_2 \overset{k_1}{\rightleftharpoons} RCO_2^- + R'NH_3^+$$

$$k_1 = \frac{(RCOO^-)\ (R'NH_3^+)}{(RCOOH)\ (R'NH_2\ )}$$

The above equilibrium relationship holds good in the absence of solvent Butyl alcohol that is used. In the presence of n-Butanol the equilibrium relation is expressed as $k_1 = \frac{k_A k_B}{k_S}$ where k_S is the ionic product of the solvent. k_A & k_B are the dissociation constants of acid and base respectively.

The rate expressions governing the mechanism discussed are:

$$-d(A)/dt = 2k_3(A)(B)(F)/(2(A)+(C)) \qquad (59)$$

$$-d(B)/dt = (B)(F)\ (2k_3(A) + k_4(C))/(2(A)+(C)) \qquad (60)$$

$$d(C)/dt = (B)(F)\ (2k_3(A) - k_4(C))/(2(A)+(C)) \qquad (61)$$

$$d(D)/dt = (B)(F)\ (k_4(C)\ /2(A)+(C)) \qquad (62)$$

where (A) = concentration of terephthalic acid in gmol/l

(B) = concentration of ethylene oxide in gmol/l

(C) = concentration of hydroxyethyl terephthalate in gmol/l

(D) = concentration of bis-hydroxy terephthalate in gmol/l

These four equations represent reactant and product distribution as a function of time.

These equations are solved for the optimum values of the parameters k_3 and k_4 which are the reaction velocity constants for the formation of HET and BHET respectively.

EXAMPLE 2

Isomerisation of m-Xylene:

The reaction scheme for the isomerisation can be represented as

(i) Series Reaction : p-Xylene $\rightleftharpoons$ m-Xylene
$\rightleftharpoons$ o-Xylene

(ii) Coupled Reaction: o-Xylene $\rightleftharpoons$ p-Xylene $\rightleftharpoons$ m-Xylene

$$1 \underset{k_2'}{\overset{k_1'}{\rightleftharpoons}} 2 \underset{k_4'}{\overset{k_3'}{\rightleftharpoons}} 3$$

p m o

k_1', k_2', k_3', k_4' are reaction rate constants.

The differential model representing this reaction mechanism would be:

$$\frac{dC_1}{dt} = -k_1' SC_1 + k_2' SC_2 \qquad (63)$$

$$\frac{dC_2}{dt} = k_1' SC_1 - k_2' SC_2 - k_3' SC_2 + k_4' SC_3 \qquad (64)$$

$$\frac{dC_3}{dt} = k_3' SC_2 - k_4' SC_3 \qquad (65)$$

where at any time t, C_1, C_2, C_3 are the concentrations of p, m, and o-Xylenes respectively and S is the active surface area of the catalyst. The rate constants k_i's are based on unit effective surface area. For the same catalyst sample used in different reactions the value of S remains unchanged and so it can be coupled with the rate constants k_i's to yield the rate equations as follows:

$$\frac{dC_1}{dt} = -k_1C_1 + k_2C_2 \qquad (66)$$

$$\frac{dC_2}{dt} = k_1C_1 - k_2C_2 - k_3C_2 + k_4C_3 \qquad (67)$$

$$\frac{dC_3}{dt} = k_3C_2 - k_4C_3 \qquad (68)$$

where $k = k'S \ hr^{-1}$.

The equations are also solved by the methods explained and discussed in the earlier chapter for the optimum value of the four reaction velocity constants k_i: $i = 1,2,3,4$.

The kinetic data used for the estimation of parameters in two different models elucidated above are:[26,27]

MODEL 1 (from Reference 26)

TABLE 2

EXPERIMENTAL CONDITIONS

Run No.	wt. of catalyst (grams)	Conc. of catalyst (gmol/l)	Temperature of the run °C	Initial TPA (grams)	Total volume (ml)
1	9.9974	0.185	60	44	535
2	2.7030	0.043	60	166	624
6	2.7159	0.0434	60	166	620
7	3.7990	0.061	100	166	620
8	9.2930	0.172	60	44	535
10	10.1699	0.160	60	166	630

TABLE 3

EXPERIMENTAL DATA

	Sample No.	Time (min)	TPA (gmol/l)	EO (gmol/l)	HET (gmol/l)	BHET (gmol/l)
Run No.1	1.	0.0	0.476	1.750	0.000	0.000
	2.	15.0	0.416	1.670	0.027	0.035
	3.	30.0	0.393	1.580	0.034	0.049
	4.	60.0	0.333	1.373	0.051	0.092
	5.	90.0	0.239	1.363	0.071	0.165
	6.	120.0	0.186	1.257	0.079	0.211
	7.	180.0	0.156	1.000	0.099	0.220
Run No.2	1.	0.0	1.590	1.545	0.000	0.000
	2.	15.0	1.570	1.543	0.007	0.006
	3.	30.0	1.540	1.518	0.010	0.037
	4.	60.0	1.530	1.395	0.010	0.049
	5.	90.0	1.510	1.318	0.010	0.056
	6.	120.0	1.470	1.300	0.012	0.068
	7.	180.0	1.460	1.268	0.017	0.074
Run No.6	1.	0.0	1.600	3.140	0.000	0.000
	2.	25.0	1.430	2.800	0.010	0.163
	3.	40.0	1.365	2.680	0.020	0.234
	4.	70.0	1.333	2.630	0.017	0.257
	5.	100.0	1.240	2.440	0.015	0.348
	6.	140.0	1.120	2.240	0.020	0.461
	7.	180.0	1.017	2.050	0.022	0.561

Contd...

	Sample No.	Time (min)	TPA (gmol/l)	EO (gmol/l)	HET (gmol/l)	BHET (gmol/l)
Run No.7	1.	0.0	1.560	1.910	0.000	0.000
	2.	30.0	1.510	1.835	0.015	0.031
	3.	55.0	1.340	1.510	0.037	0.183
	4.	70.0	1.111	1.052	0.049	0.400
	5.	100.0	0.921	0.754	0.054	0.585
	6.	130.0	0.689	0.302	0.064	0.807
	7.	180.0	0.572	0.010	0.071	0.917
Run No.8	1.	0.0	0.462	1.560	0.000	0.000
	2.	15.0	0.419	1.493	0.013	0.030
	3.	30.0	0.400	1.409	0.015	0.048
	4.	60.0	0.360	1.395	0.017	0.085
	5.	90.0	0.333	1.364	0.022	0.107
	6.	120.0	0.246	1.323	0.049	0.167
	7.	180.0	0.196	1.168	0.051	0.215
Run No.10	1.	0.0	1.500	1.560	0.000	0.000
	2.	25.0	1.470	1.510	0.005	0.024
	3.	40.0	1.450	1.448	0.011	0.042
	4.	65.0	1.410	1.395	0.017	0.072
	5.	100.0	1.272	1.240	0.019	0.209
	6.	140.0	1.166	0.900	0.011	0.323
	7.	180.0	1.106	0.800	0.017	0.377

MODEL 2 (from Reference 27)

TABLE 4

EXPERIMENTAL CONDITIONS

Data Set No.	Catalyst composition	Reaction Temperature (°K)	Catalyst loading (gm/ml)	Catalyst particle size (mm)	Stirrer speed (rpm)
1	1.5% Ni-HM	578	0.02	1.6	250
2	1.5% Ni-HM	533	0.02	1.6	250

TABLE 5

EXPERIMENTAL DATA

	Sample No.	Time (hrs.)	p-Xylene gmol/l	m-Xylene gmol/l	o-Xylene gmol/l	Toluene gmol/l	TMB gmol/l
Run No.1	1	0	0.0000	4.0000	0.0000	0.0000	0.0000
	2	1	0.1252	3.7512	0.1068	0.0072	0.0080
	3	2	0.2172	3.5700	0.1832	0.0144	0.0152
	4	4	0.3960	3.2080	0.3352	0.0304	0.0304
	5	6	0.5433	2.9068	0.4624	0.0448	0.0420
	6	8	0.6700	2.6448	0.572	0.0560	0.0572
	7	9	0.7200	2.4988	0.620	0.0808	0.0804
	8	11	0.7680	2.3580	0.704	0.0840	0.0860
	9	14	0.7736	2.3320	0.7144	0.0900	0.0900
	10	16	0.8320	2.2272	0.7680	0.0856	0.0872

Contd....

	Sample No.	Time (hrs.)	p-Xylene gmol/l	m-Xylene gmol/l	o-Xylene gmol/l	Toluene gmol/l	TMB gmol/l
Run No.2	1	1	0.0000	4.000	0.0000	0.000	0.000
	2	2	0.0380	3.938	0.0296	0.000	0.000
	3	4	0.0704	3.870	0.0600	0.000	0.000
	4	6	0.1160	3.762	0.1000	0.0108	0.0112
	5.	8	0.1650	3.668	0.1420	0.0128	0.012
	6	9	0.2360	3.532	0.1930	0.0196	0.0196
	7	11	0.2600	3.552	0.2080	0.0220	0.0216
	8	14	0.3280	3.314	0.2980	0.0248	0.0268
	9	16	0.3504	3.287	0.3060	0.0280	0.0280

CHAPTER 3

CONVERGENCE AND STABILITY OF DIFFERENTIAL EQUATIONS

3a. INITIAL ESTIMATES OF PARAMETERS:

Most of the optimisation routines require a good initial guess of parameters to attain the optimum value of the objective function. The solution may overshoot the optimum or convergence becomes excessively slower if an initial guess is quite far from the optimum. This problem would be much **more** severe in case of differential models where divergence due to poor initial estimates may get compounded with the instability and divergence due to the nonlinearity of the equations and poor step size used in integration.

A variation of linear least squares technique is used for differential models for multiresponse systems to provide near optimal initial estimates of the parameters involved in the model:

In case of linear algebraic models

$$\underline{\hat{Y}} = \underline{\underline{X}}\ \underline{\beta} + \underline{\epsilon}$$

The best linear unbiased estimate (BLUE) of parameters is given by

$$\underline{b} = (\underline{\underline{X}}^T \underline{\underline{X}})^{-1}\ \underline{\underline{X}}^T \underline{Y} \tag{69}$$

where $\underline{Y}$ is the vector of responses at different times $\underline{\underline{X}}$ is the constant matrix of independent variables and $\underline{b}$ is the BLUE of $\underline{\beta}$.

In case of differential models we have:

$$\frac{d\hat{\underline{Y}}}{dt} = \underline{\underline{X}}\ \underline{\beta}$$
$$\underline{Y} = \hat{\underline{Y}} + \underline{\epsilon} \qquad (70)$$

The vector $\underline{Y}$ at different time values is observed.

In this model, $\frac{d\hat{\underline{Y}}}{dt}$ is replaced by $\frac{\Delta\hat{\underline{Y}}}{\Delta t}$ -which is nothing but the slope of $\underline{Y}$ v/s t curve at the average value of two consecutive concentrations at different times. The design matrix $\underline{\underline{X}}$ - as it is known as - is calculated for each time interval at the average concentrations (keeping in view the mean value theorem of calculus). The RHS of Eq. (69) is evaluated in two segments : $\underline{\underline{X}}^T\underline{\underline{X}}$ and $\underline{\underline{X}}^T\underline{Y}$ and summed up over all the time intervals to form two different sum matrices respectively . The former sum matrix is inverted and multiplied with the latter to give a very closely optimal initial values of parameters:

$$\underline{b} = \left(\sum_{r=0}^{p-1} \underline{\underline{X}}^T\underline{\underline{X}}\right)^{-1} \left(\sum_{r=0}^{p-1} \underline{\underline{X}}^T \frac{\Delta\underline{Y}}{\Delta t}\right) \qquad (71)$$

where p observations are made of different reactant and product concentrations involved. A program is attached in the appendix for carrying out these calculations.

3b. STABILITY CHARACTERISTICS OF DIFFERENTIAL MODELS:

The integration methods used to solve a set of differential equations are susceptible to instability which is largely caused by a large stepsize used and, sometimes, due to highly nonlinear nature of the equations.

We will try to understand the phenomenon caused due to large step size. Let us take the test equation

$$\frac{dy}{dt} = -\lambda y \qquad y(o) = 1 \tag{72}$$

where λ is real and positive. Let us, then, write the solution as the sum of the exact solution y_{ex} and an error ϵ. We put this equation into Eq. (72) and note that the exact solution satisfies the differential equation, too. Then the error satisfies:

$$\frac{d\epsilon}{dt} = -\lambda\epsilon \tag{73}$$

We examine the error in successive time steps by looking at $\epsilon_n = \epsilon(t_n)$ and ϵ_{n+1}. An integration method is stable if the error decays in successive time steps. Because of round off errors the computer never solves equations exactly. If the scheme is unstable this round off error grows with successive time steps and soon swamps the solution.

We have used fourth order Runge-Kutta-Gill method and its variations to solve the system of nonlinear

first order initial value differential equations. So, we will calculate the maximum step size that can be used to ensure stability in the integration of our system.

The R-K-G method is expressed by the algorithm

$$
\begin{aligned}
K_1 &= h\, f\,(y_n, t_n) \\
K_2 &= h\, f\,(t_n + 1/2\, h,\; y_n + 1/2\, K_1) \\
K_3 &= h\, f\,(t_n + 1/2\, h,\; y_n + AK_1 + BK_2) \\
K_4 &= h\, f\,(t_n + h,\; y_n + CK_2 + DK_3)
\end{aligned}
\tag{74}
$$

$$y_{n+1} = y_n + 1/6\,(K_1 + K_4) + 2/6\,(BK_2 + DK_3)$$

As the analysis in case of non linear functions is carried out by linearising them around y_n, t_n, we will carry out the analysis assuming that equations are represented linearly in y.

$$\frac{dy}{dt} = -\lambda y$$

Then,

$$
\begin{aligned}
K_1 &= -\lambda h y_n \\
K_2 &= -\lambda h\,(y_n - 1/2\,\lambda\, h y_n) \\
K_3 &= -\lambda h\,\left(y_n + \left(\frac{\sqrt{2}-1}{2}\right)(-\lambda h y_n) + \left(\frac{2-\sqrt{2}}{\sqrt{2}}\right)(-\lambda h\,(y_n - 1/2\,\lambda h y_n))\right) \\
K_4 &= -\lambda h\,\left(y_n - \frac{\sqrt{2}}{2}\left(-\lambda h\left(y_n - \frac{1}{2}\lambda h y_n\right)\right) + \left(1 + \frac{\sqrt{2}}{2}\right) K_3\right)
\end{aligned}
$$

After simplification, we get :

$$K_1 = -\lambda h y_n$$

$$K_2 = -\lambda h y_n \,(1-1/2\, \lambda h)$$
$$K_3 = -\lambda h y_n \,(2- (\frac{1+\sqrt{2}}{\sqrt{2}})\ \lambda h + (\frac{\sqrt{2}-1}{\sqrt{2}}-)\ \lambda^2 h^2) \qquad (76)$$
$$K_4 = -\lambda h y_n (1+(\frac{\sqrt{2}-4-2\sqrt{2}}{2})\lambda h + (\frac{4+4\sqrt{2}-\sqrt{2}}{4})\ \lambda^2 h^2 + \frac{1}{2}\lambda^3 h^3)$$

Therefore, substituting in

$$y_{n+1} = y_n + \frac{1}{6}\ (K_1+K_4) + \frac{2}{6}\ (BK_2+DK_3) \qquad (77)$$

gives

$$y_{n+1} = y_n + \frac{y_n}{6}\ ((-8-\sqrt{2})\ \lambda h + (\ \frac{9\sqrt{2}+1}{2\sqrt{2}})\ \lambda^2 h^2 + (\frac{-12+\sqrt{2}}{4})\lambda^3 h^3$$
$$- \frac{1}{2}\ {}^4 h^4\ ) \qquad (78)$$

Let us name the second terms on R.H.S of Eq. (78) as stability term

then,

$$y_{n+1} = y_n \ (1+\frac{1}{6}\ (\text{ stability term})) \qquad (79)$$

as the exact solution satisfies the differential equations,

$$\epsilon_{n+1} = \epsilon_n\ (1+\frac{1}{6}\ (\text{stability term})) \qquad (80)$$

Now, for a method to be stable, we should have $\frac{\epsilon_{n+1}}{\epsilon_n} \leq 1$ (81)

$$\Rightarrow \quad \left|(1+\frac{1}{6}(\text{stability term}))\right| \leq 1 \qquad (82)$$

$$\Rightarrow \quad -12 \leqslant \text{stability term} \leq 0 \qquad (83)$$

Using these limits on the stability term, the term h was evaluated which turned out to be

$$|\lambda h| \leqslant 2.8 \qquad (84)$$

where λ is the maximum eigen value of the jacobian matrix $\underline{\underline{A}}$, in the equation

$$\frac{dy_i}{dt} = f_i (y_n) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial y_j} (y_j - y_{jn}) \tag{85}$$

$$\frac{\partial f}{\partial y} = \underline{\underline{A}} \text{ , the jacobian .}$$

3c. A NOTE ON STIFFNESS OF DIFFERENTIAL EQUATIONS[32]:

As we have seen in case of differential equations governed by $\frac{d\underline{y}}{dt} = \underline{\underline{A}}\,\underline{y} \quad \underline{y}(o) = \underline{G}$ (given)

$\underline{\underline{A}}$ is the matrix of coefficients

That, the largest step size in governed by $\Delta t \leq \frac{p}{|\lambda_{max}|}$ where p has different values depending upon the method and λ_{max} is the largest of the absolute magnitudes of eigenvalues.

We have, the unfortunate situation with system of equations that the largest stepsize is governed by the largest eigenvalue and the final time is unusually governed by the smallest eigen value.

We define the stiffness ratio, SR as

$$SR = \frac{\max\limits_i |\text{Re}\,\lambda_i|}{\min\limits_i |\text{Re}\,\lambda_i|} \tag{86}$$

Typically SR = 20 is not stiff, SR = 10^3 is stiff and SR= 10^6 is extremely stiff.

CHAPTER 4

RELIABILITY ANALYSIS

In comparing various algorithms for parameter estimation reliability analysis can give us the degree of confidence that can be attached to the parameter estimates.

A procedure of computing confidence regions or confidence intervals-especially suitable for differential models- is presented as described by Rosenbrock and Storey[33].

Analytical Development:

Consider two adjacent solutions $\underline{X}$ and $\underline{X} + \underline{\xi}$ of

$$\dot{\underline{X}}_i = f_i\,(\underline{X}\,, \underline{a}, t) \quad i= 1,2,\ldots,n \tag{87}$$

where $\underline{a}$ represents a p-vector of parameters.

Starting at t=0 from $\underline{X}\,(o) = \underline{C}$, $\underline{\xi}(o) = 0$ where the perturbations $\underline{\xi}$ are due to small changes $\underline{\alpha}$ in $\underline{a}^*$:

$$\dot{\underline{X}} = \underline{f}\,(\underline{X}\,, \underline{a}^*, t) \;;\; \underline{X}(o) = \underline{C} \tag{88}$$

$$\dot{\underline{X}} + \dot{\underline{\xi}} = f\,(\underline{X+\xi}\,,\ \underline{a}^* + \underline{\alpha}\,, t)\,;\ \underline{X}(o) = \underline{C} \quad \underline{\xi}\,(o) = \underline{0} \tag{89}$$

Expanding Eq.(89) in a Taylor series, and keeping only terms of the first order in $\underline{\xi}$ and $\underline{\alpha}$, we obtain

$$\dot{\underline{X}} + \dot{\underline{\xi}} = f(\underline{X}, \underline{a}^*, t) + \underline{\underline{A}}\,\underline{\xi} + \underline{\underline{B}}\,\underline{\alpha} \tag{89a}$$

$$\dot{\underline{\xi}} = \underline{\underline{A}}\,\underline{\xi} + \underline{\underline{B}}\,\underline{\alpha} \;;\; \underline{\xi}\,(o) = \underline{0}$$

where $\underline{\underline{A}} = (A_{ij}) = \frac{\partial}{\partial x_j} f_i\ (\underline{X}, \underline{a}^*, t)$

$\underline{\underline{B}} = (B_{ij}) = \frac{\partial}{\partial a_j} f_i\ (\underline{X}, \underline{a}^*, t)$

the matrix A is nxn and B is nxp; both matrices are functions of t but not of $\underline{\xi}$ or $\underline{\alpha}$.

Now, consider the matrix differential equation

$$\dot{\underline{\underline{Y}}} = -\underline{\underline{A}}'\underline{\underline{Y}}; \quad \underline{\underline{Y}}(o) = \underline{\underline{I}}$$

where $\underline{\underline{Y}}$ is an nxn matrix and $\underline{\underline{I}}$ is the identity matrix. We have

$$\frac{d}{dt}\ \underline{\underline{Y}}'\ \underline{\xi} = \dot{\underline{\underline{Y}}}'\ \underline{\xi} + \underline{\underline{Y}}'\ \dot{\underline{\xi}}$$

$$= -\underline{\underline{Y}}'\underline{\underline{A}}\underline{\xi} + \underline{\underline{Y}}'(\underline{\underline{A}}\ \underline{\xi} + \underline{\underline{B}}\ \underline{\alpha}) = \underline{\underline{Y}}'\underline{\underline{B}}\ \underline{\alpha}$$

cancellation of terms containing $\underline{\xi}$ is the motive for introducing the adjoint equations. Then since $\underline{\xi}(o) = \underline{0}$

$$\underline{\underline{Y}}'(t_r)\ \underline{\xi}\ (t_r) = \int_o^{t_r} \underline{\underline{Y}}'(t)\ \underline{\underline{B}}(t)\ \underline{\alpha}\, dt \qquad (r=1,2,..,n)$$

$$= \underline{\underline{C}}\ (t_r)\ \underline{\alpha}$$

where nxp matrix $\underline{\underline{C}}(t_r)$ is $\quad \underline{\underline{C}}(t_r) = \int_o^{t_r} \underline{\underline{Y}}'(t)\ \underline{\underline{B}}(t)\ dt$

The matrix $\underline{\underline{Y}}'(t_r)$ is assumed to be nonsingular for all t_r, and then we have: $\underline{\xi}\ (t_r) = (\underline{\underline{Y}}'(t_r))^{-1}\ \underline{\underline{C}}(t_r)\ \underline{\alpha} = \underline{\underline{D}}\ (t_r)\ \underline{\alpha}$ say where $\underline{\underline{D}}_r$ is nxp. If follows that

$$\underline{\underline{D}}_r = (D_{ij}(t_r)) = \frac{\partial}{\partial a_j}\ X_i\ (\underline{a}^*, t_r))$$

Thus, $\underline{\underline{D}}_r$ gives the dependence of $\underline{X}(t_r)$ upon variations $\underline{\alpha}$ of $\underline{a}$ around $\underline{a}^*$ which is what we set out to find.

Consider next the function $F(\underline{a}^* + \underline{\alpha})$ as given by

$$F(\underline{a}) = \sum_{r=1}^{R} \left\{ \underline{\underline{Y}}'(t_r) - \underline{g}'(\underline{X}(a, t_r)) \right\} \underline{\underline{W}}_r \left\{ \underline{\underline{Y}}(t_r) - \underline{\underline{g}}(X(a, t_r)) \right\}$$

Let $\underline{\alpha}$ be chosen to make $F(\underline{a}^* + \underline{\alpha})$ a minimum, so that $\underline{a}^* + \underline{\alpha} = \bar{a}$.

Then for $i = 1, 2, \ldots, p.$

$$\frac{\partial}{\partial \alpha_i} \sum_{r=1}^{R} \left\{ (\underline{\underline{Y}}'(t_r) - \underline{\underline{g}}'(\underline{X}(\underline{a}^* + \underline{\alpha}, t_r)) \underline{\underline{W}}_r (\underline{\underline{Y}}(t_r) - \underline{\underline{g}} (\underline{X}(\underline{a}^* + \underline{\alpha}, t_r)) \right\} = 0$$

or

$$\frac{\partial}{\partial \alpha_i} \sum_{r=1}^{R} (g_r' + \eta_r' - g_r' - \alpha' D' G_r')(\underline{\underline{W}}_r (g_r + \eta_r - g_r - G_r D_r \alpha)) = 0$$

where $g_r = g(X(a^*, t_r))$

$$\underline{\underline{G}}_r = (G_{ij})_r = \left(\frac{\partial}{\partial x_j} g_i (X(\underline{a}^*, t_r)) \right)$$

and $\underline{\underline{D}}_r$ is given by Eq. $\left(\frac{\partial}{\partial a_j} x_i(a^*, t_r) \right)$

The matrix $\underline{\underline{G}}_r$ is mxn, D_r is nxp.

Thus, we have

$$\sum_{r=1}^{R} \underline{\underline{D}}_r' \underline{\underline{G}} \underline{\underline{W}}_r (\eta_r' - \underline{\underline{G}}_r \underline{\underline{D}}_r \underline{\alpha}) = 0$$

$$\left(\sum_{r=1}^{R} \underline{\underline{D}}_r' \underline{\underline{G}}_r' \underline{\underline{W}}_r \underline{\underline{G}}_r \underline{\underline{D}}_r \right) \underline{\alpha} = \sum_{r=1}^{R} \underline{\underline{D}}_r' \underline{\underline{G}}_r' \underline{\underline{W}}_r \underline{\eta}_r$$

or

$$\underline{\underline{H}} \; \underline{\alpha} = \sum_{r=1}^{R} \underline{\underline{D}}_r' \underline{\underline{G}}_r' \underline{\underline{W}}_r \underline{\eta}_r \qquad (90)$$

If the symmetric matrix $\underline{\underline{H}}$ is nonsingular above Eq. determines $\underline{\alpha}$ from the $\underline{\eta}_r$. The expectation of $\underline{\alpha}$ will be zero, and we are interested in the covariance of the elements of $\underline{\alpha}$; that is, in the expected value P of $\underline{\alpha}\underline{\alpha}'$. By Eq. (90)

$$E(\underline{\underline{H}}\ \underline{\alpha}\underline{\alpha}'\underline{\underline{H}}) = E(\sum_{r=1}^{R}\sum_{r=1}^{R} \underline{\underline{D}}'_r\ \underline{\underline{G}}'_r\ \underline{\underline{W}}_r\ \underline{n}_r\ \underline{n}\ \underline{\underline{W}}'_s\ \underline{\underline{G}}_s\ \underline{\underline{D}}_r)$$

Whence by using $E(\underline{n}_r\ \underline{n}'_s) = 0 \quad r \neq s$ and $E(\underline{n}_r\ \underline{n}'_r) = \underline{\underline{M}}_r$ we have

$$\underline{\underline{H}}\ \underline{\underline{P}}\ \underline{\underline{H}} = \sum_{r=1}^{R} \underline{\underline{D}}'_r\ \underline{\underline{G}}'_r\underline{\underline{W}}_r\underline{\underline{M}}_r\ \underline{\underline{W}}_r\ \underline{\underline{G}}_r\ \underline{\underline{D}}_r$$

This equation determines P if H is nonsingular . By Eq. (90) $\underline{\alpha}$ is a linear function of $\underline{n}_r$ so that its probability density will be Gaussian and will be given by:

$$\frac{1}{(2\pi)^{p/2}\sqrt{|\underline{\underline{P}}|}}\ \exp\left(-\frac{1}{2}\underline{\alpha}'\ \underline{\underline{P}}^{-1}\underline{\alpha}\right)$$

If now $\underline{b}$ is a unit p-vector the linear function $\underline{b}'\underline{\alpha}$ of $\underline{\alpha}$ will have Gaussan distribution with variance

$$\sigma_b^2 = \underline{b}'\ \underline{\underline{P}}\ \underline{b}$$

The confidence interval $\bar{r}(\underline{b})$ associated with $\underline{b}'\bar{a}$ is then

$$r(\underline{b}) = K\sigma_b \qquad (92)$$

All the equations used sofar can be put into a computationally convenient form

$$\dot{\underline{X}} = f(\underline{X}, \underline{a}, t); \quad \underline{X}(o) = \underline{C} \tag{93}$$

$$\bar{a} : \min_{a} \sum_{r=1}^{R} \left\{ \underline{Y}'(t_r) - \underline{g}'(X(\underline{a}, t_r)) \right\} M_r^{-1} \left\{ \underline{Y}(t_r) - \underline{g}(X(\underline{a}, t_r)) \right\} \tag{94}$$

$$\underline{\underline{A}} = (A_{ij}) = \left(\frac{\partial}{\partial x_j} f_i (\underline{X}, \bar{a}, t)\right) \tag{95}$$

$$\underline{\underline{B}} = (B_{ij}) = \left(\frac{\partial}{\partial a_j} f_i (\underline{X}, \bar{a}, t)\right) \tag{96}$$

$$\dot{\underline{\underline{D}}} = \underline{\underline{A}}(t)\, \underline{\underline{D}} + \underline{\underline{B}}(t) \quad ; \underline{\underline{D}}(o) = 0 \tag{97}$$

$$\underline{\underline{D}}_r = \underline{\underline{D}}(t_r) \tag{98}$$

$$\underline{\underline{G}}_r = \underline{\underline{G}}(t_r) = (G_{ij}(t_r)) = \frac{\partial}{\partial x_j} g_i(X(\bar{a}, t_r))) \tag{99}$$

$$\underline{\underline{H}} = \sum_{r=1}^{R} \underline{\underline{D}}_r' \, \underline{\underline{G}}_r' \, \underline{\underline{M}}_r^{-1} \, \underline{\underline{G}}_r \, \underline{\underline{D}}_r \tag{100}$$

$$\underline{\underline{P}} = \underline{\underline{H}}^{-1} \tag{101}$$

$$\sigma_b^2 = b' \, \underline{\underline{P}} \, b \tag{102}$$

In applying these formulae , it is assumed that $\bar{a}$ is found by using standard optimisation methods. This will involve many successive integrations of Eq. (93) with different values of $\bar{\underline{a}}$.

When $\bar{a}$ has been found it becomes possible to estimate σ_b and so to find the confidence intervals $r(\underline{b})$. For this purpose a further integration of Eq. (93) is begun, using $\underline{a} = \bar{\underline{a}}$. At each step of integration $\underline{\underline{A}}$ and $\underline{\underline{B}}$ are evaluated: Thus, the solutions of Eqs. (93) and (97) is carried out simultaneously, and $\underline{\underline{D}}$ is therefore available

whenever it is required. Since $\underline{X}$ is available $\underline{\underline{G}}$ can also be computed whenever it is needed.

At each time t_r, when an observation is available, the value of $\underline{\underline{D}}_r' \underline{\underline{G}}_r' \underline{\underline{M}}_r^{-1} \underline{\underline{G}}_r \underline{\underline{D}}_r$ can be computed and added to the running sum which forms $\underline{\underline{H}}$. Once $\underline{\underline{H}}$ matrix is obtained, by inversion $\underline{\underline{P}}$ can be obtained or analysis can be carried out by finding eigenvalues and eigenvectors of $\underline{\underline{H}}$; Eigenvalues of $\underline{\underline{P}}$ are reciprocal of eigenvalues of $\underline{\underline{H}}$ and eigenvectors are same. These eigenvalues and eigenvectors reveal the situation completely.

4a. EXAMINATION OF RESIDUALS:

Residual analysis is useful and valid not only for the linear regression models but also for nonlinear regression models and other forms of models.

The residuals are defined as the n differences $e_i = Y_i - \hat{Y}_i$, i= 1,2..,n . Where Y_i is an observation and $\hat{Y}_i$ is the corresponding fitted value obtained by using proposed model equations.

Now while performing the regression analysis we have made certain assumptions about the errors; the usual assumptions are that the errors are independent, have zero mean, a constant variance σ^2 , and follow a normal distribution. Thus, if our fitted model is correct, the residuals should exibit

tendencies that tend to confirm the assumptions that we have made, or atleast, should not exibit a denial of the assumptions. After the examination of residuals we can conclude that either

1. The assumptions appear to be violated (in a way that can be specified), or
2. The assumptions do not appear to be violated. The ways to examine residuals are graphical as well as numerical and are easy to do; they are very revealing when the assumptions are violated. The principal ways of plotting the residuals e_i are

   1. Overall
   2. In time sequence; if the order is known.
   3. Against the fitted values $\hat{Y}_i$

Statistics for examination of residuals:

The plots we have discussed above are visual tests to check some of the basic assumptions underlying the estimation of parameters. We may, however, use statistics to do the same analysis:-

$$T_{pq} = \sum_{i=1}^{n} e_i^{p} \hat{Y}_i^{q} \qquad (103)$$

$$T_{21} = \sum_{i=1}^{n} e_i^{2} \hat{Y}_i \qquad (104)$$

provides a measure of the defect of changing variance in time sequence .

$$T_{11} = \sum_{i=1}^{n} e_i \hat{y}_i \quad \text{This should always be zero} \qquad (105)$$

$$T_{12} = \sum_{e_i=1}^{n} e_i \hat{Y}_i^{2} \quad \text{provides a measure of defect}$$

due to the absence of linear and quadratic terms of time in the model.

CHAPTER 5

RESULTS AND DISCUSSIONS:

A comparison of four different algorithms to estimate parameters in models governed by ordinary differential equations is carried out so as to help the future investigators in selecting a method for parameter estimation in kinetic studies involving dynamic responses.

NUMERICAL PROCEDURE:

The four algorithms discussed in chapter 2 were applied on the experimental data presented in Tables 2-4.

In the first algorithm we require initial conditions to start the integration of the equations. Initial values of the parameters are also required. The step by step procedure is given below:

(1) Model Used:

(a) $d(A)/dt = - 2\ k_3(A)(B)(F)/\ (2(A)+(C))$

(b) $d(B)/dt = - (B)(F)\ (2k_3(A)+K_4(C)/(2(A)+(C))$

(c) $d(C)/dt = - (B)(F)(2k_3(A)-K_4(C))/(2(A)+(C))$

(d) $d(D)/dt = - (B)(F)(K_4(C)/2(A)\ +(C))$

where the symbols have the meaning as given in chapter 2.

TABLE 5

Run No.1

Initial conditions are (gmol/l)	Initial values of the parameters $(mol/l)^{-1} \frac{1}{min}$
(A) = 0.476	K_3 = 0.0107
(B) = 1.750	K_4 = 0.0497
(C) = 0.000	
(D) = 0.000	
(F) = 0.185 (Constant for the entire run)	

Run No.2

(A) = 1.590	K_3 = 0.0092
(B) = 1.545	K_4 = 0.0762
(C) = 0.000	
(D) = 0.000	
(F) = 0.043	

Run No.6

(A) = 1.600	K_3 = 0.0066
(B) = 3.140	K_4 = 0.1170
(C) = 0.000	
(D) = 0.000	
(F) = 0.0434	

Run No.7

Initial conditions are (gmol/l)	Initial values of the parameters $(mol/l)^{-1}\ \frac{1}{min}$
(A) = 1.560	K_3 = 0.1310
(B) = 1.910	K_4 = 3.2240
(C) = 0.000	
(D) = 0.000	
(F) = 0.061	

Run No.8

(A) = 0.462	K_3 = 0.0072
(B) = 1.560	K_4 = 0.1670
(C) = 0.000	
(D) = 0.000	
(F) = 0.172	

Run No.10

(A) = 1.500	K_3 = 0.0102
(B) = 1.560	K_4 = 0.0720
(C) = 0.000	
(D) = 0.000	
(F) = 0.160	

The R.H.S. of Eqns. (a) - (d) is evaluated at the given conditions with the parameter values as mentioned above. This function is used as f ($\underline{Y}$) in the algorithm and the R-K-G parameters K_1,K_2,K_3 and K_4 are evaluated. Using

these constants, we can get the concentrations at the next time value. The time interval between two consecutive observations is used as the stepsize in the solution of the equations. Once the concentrations of A,B,C and D, as predicted by the model are available, these are subtracted from the observed values reported in the data tables. These residuals are then squared and added to a running sum to give the value of the objective function for the minimization program. Then the minimization program carries out search for the minimum and the parameter values are changed. Again starting from the same initial conditions, the above procedure is repeated to give the next measure of the objective function which is again checked for the minimum. This way the search continues for the minimum objective function and once it is reached, the corresponding values of the parameters provide the optimal estimates. A program listing is included in the appendix.

In algorithm 2 , a very similar procedure is employed till the calculation of the residuals, then the variance- covariance matrix of these residuals is constructed. The procedure is discussed in chapter 2. Determinant of this matrix is found out by employing complete pivoting technique as given in the program at the end . This determinant serves as the objective function and is fed into the complex minimization routine. The parameter values are updated and again

starting from the initial conditions, the objective function is evaluated till a local optimum is reached by the program. The values of the parameters corresponding to this situation are reported in the tables 6.

In algorithm 3, the equations are solved implicitly. A jacobian matrix Jac is constructed first. This is the matrix of derivatives of the functions on the R.H.S. of the equations. This matrix is multiplied by o.5h which is half the difference in the two successive time values. Now this matrix is evaluated at the initial conditions given above and then subtracted from an identity matrix of the same order, $\underline{\underline{I}}$ as discussed in the chapter 2. The resulting matrix is inverted and multiplied by the matrix $\underline{\underline{F}}$ $(\underline{Y},\underline{\theta})$ which is obtained by substituting initial conditions and initial parameter values. This, is then multiplied by h again. The result is K_1 vector which is added to the initial conditions to give the concentrations A,B,C and D at the next time value. This way the calculations are performed for all the time intervals starting from 0 to 180 minutes. These predicted values are subtracted from the observed values and then the residual sum of squares evaluated. This is fed into the nonlinear least squares program as objective function. The resulting parameter values are then used along with the same initial conditions and the procedure repeated until optimum is reached. The optimum

is reached.
parameter values are reported. A detailed program is attached in the appendix.

In algorithm 4, matrix $\underline{\underline{A}}$ is first evaluated. This matrix is has concentration terms as the elements. The matrix for the models used is given below:

Model (1)

$$\underline{\underline{A}} = \begin{bmatrix} -\frac{2(A)(B)(F)}{2(A)+(C)} & 0.0 \\ -\frac{2(A)(B)(F)}{2(A)+(C)} & -\frac{(B)(C)(F)}{2(A)+(C)} \\ -\frac{2(A)(B)(F)}{2(A)+(C)} & -\frac{(B)(C)(F)}{2(A)+(C)} \\ 0.0 & \frac{(B)(C)(F)}{2(A)+(C)} \end{bmatrix}$$

Model (2)

$$\underline{\underline{A}} = \begin{bmatrix} -C_1 & C_2 & 0.0 & 0.0 \\ C1 & -C_2 & -C_2 & C_3 \\ 0.0 & 0.0 & C_2 & -C_3 \end{bmatrix}$$

Average concentrations for each time interval are found out by taking arithmatic average of the observed values. These concentrations are used to evaluate $\underline{\underline{A}}$ matrix. $A^T A$ is found out and then multiplied by a scalar Δt_r , the corresponding time interval. This matrix is determined for all the time intervals and added to a sum matrix. The sum matrix is then inverted. Separately, $\underline{\underline{A}}^T$ matrix is evaluated and

multiplied by a vector ΔY_r which is nothing but the difference vector of two successive observed concentration values for the time interval, r used earlier. This vector is summed up over all the time intervals and then multiplied to the earlier inverted matrix. This directly gives the vector of parameters as outline in chapter 2. The algorithms are repeated for the model 2 also using the same programs. All the results are summarised in tables 6-8.

The methods were compared for their parameter values, execution time required on a DEC 1090 computer, and degree of reliability that can be attached with the parameter estimates. Approximate analytical solutions were also used to get alternate estimates of parameters which may serve as references.

A variation of linear least squares estimation was used for differential equations as outlined in chapter 2 earlier , to provide the initial estimates of parameters so that convergence can be achieved in the program for minimisation of objective function. The method worked satisfactorily for the both the models tested and gave estimates that were quite close to the optimal values that were found after the minimisation was accomplished. The method can be safely used in case of exigency or lack of good computer facilities.

After the initial estimates were found, parameters were evaluated using four different algorithms out of which three employed minimisation and the fourth evaluated parameters directly from the data. It was found that Weighted Residual Method gives the best estimates out of all the four methods, tested as was reglected by the reliability estimates- which are nothing but the confidence limits for the parameters- and the residual analysis carried out; the parameters obtained from this method gave the least objective function value showing the closeness of predicted value of multiresponses to the corresponding observed ones. The method also required a negligible central processing time of 0.11 seconds on the DEC 1090 installation . The equations representing Model 1 were nonlinear in nature and did pose some divergence problems in the integration step and, therefore, it is quite beneficial to use a method that can circumvent this step as is done in the Weighted Residual Method .

MODEL 1

TABLE 6a

INITIAL ESTIMATE OF PARAMETERS

Run No.	Initial Conc. of TPA (gmol/l)	Initial Conc. of EO (gmol/l)	rate constant k_3 for HET (gmol/l min)	rate constant for BHET k_4 ($\frac{gmol}{l\ min}$)
1.	0.476	1.750	0.0107	0.0437
2.	1.590	1.545	0.0092	0.0762
5.	1.600	3.140	0.0066	0.1170
7.	1.560	1.910	0.1310	3.2240
8.	0.462	1.560	0.0072	0.1670
10.	1.500	1.560	0.0102	0.0720

MODEL 2

TABLE 6b

INITIAL ESTIMATES OF PARAMETERS

Run No.	Initial conc. of m-xylene (gmol/l)	Rate constant (hr^{-1}) K_1	K_2	K_3	K_4
1.	4.000 (578°K)	0.0715	0.03148	0.0362	0.0962
2.	4.000 (533°K)	0.0099	0.00594	0.00725	0.0160

MODEL 1

TABLE 7a

COMPARISON OF PARAMETER ESTIMATES OBTAINED FROM DIFFERENT ALGORITHMS

Run No	Measures of comparison	Fourth order R-K-G with nonlinear ls minimization.	Fourth order R-K-G with Box complex minimisation	Implicit one point collocation with N.L.L.S. minimisation	Weighted Residual Method
1.	PARAMETERS				
	K_3	0.0090	0.0095	0.0091	0.0099
	K_4	0.0781	0.0567	0.0783	0.0491
2.	CPU TIME (Sec)	0.67	0.69	0.74	0.11
3.	RELIABILITY (K_3,K_4 $\pm$ Std. deviation)	0.11250	0.08650	0.110	0.04376

TABLE 7a.2

	MEASURES OF COMPARISON	Fourth Order R-K-G with nonlinear ls minimization.	Fourth order R-K-G with Box complex minimisation.	Implicit one point collocation with N.L.L.S Minimisation	Weighted Residual Method.
Run No.2	PARAMETERS				
	K_3	0.008	0.0076	0.0077	0.0082
	K_4	0.0845	0.0762	0.0820	0.0714
	CPU TIME (SEC)	0.67	0.69	0.74	0.11
	RELIABILITY (K_3,K_4± Std. Deviation)	0.0925	0.0872	0.0791	0.0670
Run No.6	PARAMETERS				
	K_3	0.0081	0.0077	0.0075	0.0074
	K_4	0.1120	0.0954	0.1024	0.1050
	CPU TIME (SEC)	0.67	0.69	0.74	0.11
	RELIABILITY (K_3,K_4± Std. deviation)	0.0826	0.0769	0.0656	0.0542

	PARAMETERS				
	K_3	0.142	0.136	0.132	0.139
	K_4	3.224	3.046	3.219	3.212
	CPU TIME (SEC)	0.68	0.69	0.75	0.11
Run No.7	RELIABILITY (K_3,K_4± Std. deviation)	0.1255	0.1134	0.1134	0.1044
	PARAMETERS				
	K_3	0.0066	0.0068	0.0071	0.0072
	K_4	0.1320	0.1350	0.1220	0.1290
Run No.8	CPU TIME (SEC)	0.67	0.69	0.74	0.11
	RELIABILITY (K_3,K_4± Std. deviation)	0.0944	0.0872	0.0675	0.0594
	PARAMETERS				
	K_3	0.0089	0.0086	0.0084	0.0082
	K_4	0.0982	0.0950	0.0912	0.0925
Run No.10	CPU TIME (SEC)	0.67	0.69	0.73	0.11
	RELIABILITY (K_3,K_4± Std. deviation)	0.1122	0.0989	0.0745	0.0674

MODEL 2

TABLE 7b

COMPARISON OF PARAMETER ESTIMATES OBTAINED FROM DIFFERENT ALGORITHMS

Run No.1	MEASURES OF COMPARISON	Fourth Order R-K-G with nonlinear ls minimization.	Fourth order R-K-G with Box Complex minimisation.	Implicit one point collocation with N.L.L.S Minimisation	Weighted Residual Method.	Analytical solutions
1.	PARAMETERS (hr^{-1})					
	K_1	0.0761	0.0763	0.0726	0.0717	0.0721
	K_2	0.0362	0.0315	0.0293	0.0324	0.0344
	K_3	0.0299	0.0321	0.0351	0.0371	0.0491
	K_4	0.0594	0.0645	0.0772	0.0844	0.0665
2.	CPU TIME (SEC)	0.6600	0.6800	0.7200	0.1000	0.1800
3.	RELIABILITY OF ESTIMATES	0.05715	0.05421	0.05405	0.0484	0.0523
Run No.2						
1.	PARAMETERS (hr^{-1})					
	K_1	0.0091	0.0094	0.0092	0.0099	0.0097
	K_2	0.0072	0.0063	0.0067	0.0062	0.0058
	K_3	0.0067	0.0077	0.0074	0.0074	0.0071
	K_4	0.0135	0.0146	0.0104	0.0152	0.0164
2.	CPU TIME (SEC)	0.6700	0.6900	0.7300	0.1100	0.1800
	RELIABLITY OF ESTIMATES	0.0324	0.0288	0.0307	0.0072	0.0084

MODEL 1

TABLE 8

COMPARISON OF R.S.S. WITH DET (Σ) IN ALGORITHM USING BOX-COMPLEX METHOD

Run No.	RESIDUAL SUM OF SQUARES AS OBJECTIVE FUNCTION	DET (Σ), AS OBJECTIVE FUNCTION*
1.	0.0754	0.52×10^{-8}
2.	0.0682	0.47×10^{-8}
6.	0.0722	0.44×10^{-8}
8.	0.0541	0.28×10^{-8}

* This objective function was used in the estimation of parameters while using algorithm 2.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Parameter estimation in case of ordinary differential equations is studied, in detail:

1. The existing literature on the subject is reviewed and the importance of the subject is stressed.
2. Four different methods which cover the different ways in which the problem is tackled in the literature are compared.
3. To provide the kinetic investigator with a good approximation of the parameters involved in his differential models, a variation of linear least squares estimation is proposed.
4. Weighted Residual Methods are advocated for the purpose, for the advantage of speedy accuracy and ease of handling the equations.
5. Residual analysis is carried out for both the test models to check the adequacy of error distribution assumptions. Nothing concrete can be said by looking at these plots.
6. Computer programs are developed which treat the subject of parameter estimation for the case of differential models with good accuracy and reliability.

The field of parameter estimation and model discrimination has offered an enormous potential for research in the last couple of decades and, due to, the pioneering

work of some of the workers progressed very rapidly to its culmination.

The future work in this direction could be aimed at the successful use of these techniques as a tool in the kinetic investigation of industrially important reactions. In case of dynamic models, the error variance is not constant [34] and therefore, parameter estimation and model discrimination based on conventional techniques may give erroneous results and therefore, the algorithms ~~thmx~~ involving integration of differential equations may be modified to account for changing variance of the errors. More so, in case of complex reaction networks where one is faced with a range of products and reactants and resorts to integral reactor for kinetic investigation due to higher conversions required. The experimental procedure could be sequentially designed to save alot of time in experimentation by cutting down the number of runs required and the tremendous human effort that is wasted in carrying these experiments without any plan. Thus, a practical use of these techniques is more desired rather than some abstract work in this area which will only amount to wastage of computer time.

BIBLIOGRAPHY

1. Himmelblau, D.M. " Process Analysis by Statistical Methods" John Wiley & Sons. Inc. 1970.
2. Frankaerts, J. and Froment, G.F., Chem.Eng. Sci., 19,807 (1964)
3. Hosten, L.H., Froment, G.F., Proc. 4th international/6th European Chemical reaction Engineering Sym. Heidelberg, 6-8 April, 1976.
4. Davies, O.L., " Design of Experiments" ELBS (1970).
5. Kittrel, J.R., Mezaki,R., Watson, C.C., Ind. Eng. Chem. 58,51 (1966).
6. Blakemore, J.W. and Hoerl, A.E., Chem.Eng. Progr.Symp.Ser. 59, No.42,14 (1963)
7. Levenspiel, O., " Chemical Reaction Engineering", Wiley, NY (1962).
8. Rubin, D.I., Chem.Eng. Progr. Symp. Ser. 59, No.42, 90 (1963).
9. Wright, B.S., Brit. Chem.Eng. 9, 758 (1964)
10. Garfinkel, D., Rutledge, J.D., Higgins, J. Comm.Assoc. Computing Machinary. 4, 559 (1961).
11. Lindsay, K.L., Ind. Eng. Chem. Funda. 1 , 241 (1962).
12. Steiner, R., and Shoennemann, K., Chem. Ing. Tech. 37,101 (1965).
13. Bak,K. Acta Chem.Sc. and 17, 985 (1963).
14. Ball, W.E. and Groenweghe, L.C.D., Ind. Eng. Chem.Funda. 5, 181 (1966).

15. Cull, N.I. and Brenner, H.H., Ind. Eng. Chem. 53, 833(1961).

16. Peterson, T.I., Chem. Eng. Progr. Symp. Ser. 59, No.31, 111 (1960).

17. Perterson, T.I., Chem. Eng. Sci. 17, 203 (1962).

18. Katarov, V.V and Lutsenko, V.A, Intern. Chem.Eng. 5, 623 (1965).

19. Schrodt V.N., Frank, L.E., Parisot, P.E., American Chemical Society meeting, Houston, TX. DEC (1963).

20. Howland, J.L. and Vaillancourt, R., J. Soc.Ind. Appl. Math.9 , 165 (1961).

21. Bellman,R. and Kalaba, R., " Quasilinearisation and Nonlinear Boundary Value problems", American Elsevier (1965).

22. Box, G.E.P. and Hunter, W.G., Technometrics 4, 301 (1962).

23. Himmelblau, D.M., Jones,C.R., Bischoff, K.B. Ind. Eng.Chem. Funda. 6, 543 (1967).

24. Cresswell, D.L. and Paterson, W.R., DECHEMA, Heidelberg (1976).

25. Draper, N.R. and Simth, H., " Applied Regression Analysis", Wiley, NY (1966).

26. Bhatia, S., Ph.D. Thesis I.I.T. Kanpur (1976).

27. Sensharma, P.K., M.Tech. Thesis I.I.T. Kanpur (1983).

28. Gill, P.E. and Murray, W., SIAM J.Num. Anal 15 (5),977 (1978).

29. Box, M.J., Computer Journal 8, 42 (1966).

30. Spendley, W, Hext, G., Himsworth,F., Technometrics 4,11 (1962).

31. Box, G.E.P. Anals of New York Academy of Sciences (1960).

32. Villadsen and Michelson, " Polynomial Approximation Methods in Differential Equation Models"

33. Rosenbrock and Storey " Computational techniques for Chemical Engineers " AP (1966).

34. Kittrell, J.R., Hunter, W.J., Watson,C.C., AIChEJ 11, 1051 (1965).

35. Ayens, R., J. and Peters, M.S., Ind. Eng. Chem.Process Design & Development 1 ,204 (1962).

36. Hall, G. and Watt, J.M. (Editors) " Modern Numerical Methods for Ordinary Differential Equations, Oxford Univ. Press, (1976).

37. Hosten, L.H. and Emig, G., Chem. Eng. Sci. 30, 1357 (1975).

38. Box, G.E.P. and Lucas, H.L., Biometrika 46, 77 (1959).

39. Hosten, L.H., Chem. Eng. Sci. 29, 2247 (1974).

```
C      MAIN PROGRAM TO CARRY OUT PARAMETER ESTIMATION IN MODELS
C      GOVERNED BY ORDINARY DIFFERENTIAL EQUATIONS.
C      ****************************************************************
C
C      PROGRAM SEGMENT 1. :INITIAL ESTIMATES
C      *********************************
C      THIS PROGRAM FINDS THE INITIAL ESTIMATES OF PARAMETERS IN MODE
C      GOVERNED BY FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS
C
       REAL Y(3,10),AVRGY(3,9),TIME(10),X(3,4),XT(4,3),XTX(4,4),XTY(4
      1),RATE(3,9),SUM1(4,4),SUM2(4),SUMIN(4,4),THETA(4),WKSPCE(18),
      2IDEN(4,4),AA(4,4),BB(4,4),AVCON(3),RATE1(3)
       DATA ((IDEN(I,J),I=1,4),J=1,4)/1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0
      1.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0/
       DATA ((SUM1(I,J),I=1,4),J=1,4)/16*0.0/,(SUM2(I),I=1,4)/4*0.0/
       OPEN(UNIT=21,DEVICE='DSK',FILE='FOR21.DAT')
       OPEN(UNIT=40,DEVICE='DSK',FILE='FOR40.DAT')
       READ(21,*) ITIME,ICON,IPARA,CATLST
       READ(21,*) ((Y(I,J),I=1,ICON),J=1,ITIME)
       READ(21,*) (TIME(I),I=1,ITIME)
C      ECHO BACK THE VALUES READ
       WRITE(40,999) ITIME,ICON,IPARA,CATLST
       WRITE(40,998) ((Y(I,J),I=1,ICON),J=1,ITIME)
       WRITE(40,997) (TIME(I),I=1,ITIME)
999    FORMAT(5X,'NO. OF MEASUREMENTS=',I3//5X,'NO. OF SPECIES=',I3//
      15X,'NO. OF PARAMETERS=',I3//5X,'CATALYST CONCENTRATION FOR TH
      1RUN=',F6.4,'gmmoles/litre'//)
C      FORMAT TO BE CHANGED FOR DIFFERENT NO. OF SPECIES
998    FORMAT(5X,'THE CONCENTRATION OBSERVATION MATRIX IS'//(3(5X,F12
      1))//)
997    FORMAT(5X,'THE TIME IN HRS AT WHICH THE OBSERVATIONS WERE MADE
      1/(5X,F6.2))
C      ACTUAL CALCULATIONS START FROM HERE-----------------------------
       ITIME1=ITIME-1
       CALL SLOPE(Y,TIME,RATE,ITIME,ITIME1,ICON)
       CALL AVRAGE(Y,AVRGY,ICON,ITIME,ITIME1)
C      CALCULATIONS ARE DONE FOR EACH TIME INTERVAL AND SUMMED UP---
       DO 100 I=1,ITIME1
       DO 10 J=1,ICON
       AVCON(J)=AVRGY(J,I)
       RATE1(J)=RATE(J,I)
10     CONTINUE
       CALL DMAT(X,AVCON,CATLST,ICON,IPARA)
       CALL TRNMAT(X,XT,ICON,IPARA)
       CALL MATMUT(XT,X,XTX,IPARA,ICON,IPARA)
       CALL MATMT(XT,RATE1,XTY,IPARA,ICON)
       DO 20 KK=1,IPARA
       DO 20 JJ=1,IPARA
       SUM1(KK,JJ)=SUM1(KK,JJ)+XTX(KK,JJ)
       SUM2(KK)=SUM2(KK)+XTY(KK,1)
20     CONTINUE
100    CONTINUE
       TYPE*,((SUM1(I,J),I=1,IPARA),J=1,IPARA)
C      PARAMETERS FOR INVERSION ROUTINE;CHANGE WITH MODEL
       N=IPARA
       M=IPARA
       IA=IPARA
       IB=IPARA
       IC=IPARA
       IAA=IPARA
```

```
      IBB=IPARA
      IFAIL=0
      CALL F04AEF(SUM1,IA,IDEN,IB,N,M,SUMIN,IC,WKSPCE,AA,IAA,BB,IBB
     1AIL)
      CALL MATMT(SUMIN,SUM2,THETA,IPARA,IPARA)
      WRITE(40,996) (THETA(I),I=1,IPARA)
996   FORMAT(5X,'THE INITIAL ESTIMATES OF PARAMETERS ARE'//(5X,F10.
      STOP
      END
C
C     FIRST SUBROUTINE TO FIND RATE OF REACTION
C     ***************************************************************
      SUBROUTINE SLOPE(Y,TIME,RATE,MM,M,N)
      DIMENSION Y(N,MM),TIME(MM),RATE(N,M)
      DO 10 I=1,N
      DO 10 J=1,M
      DELY=Y(I,J+1)-Y(I,J)
      DELT=TIME(J+1)-TIME(J)
      RATE(I,J)=DELY/DELT
10    CONTINUE
      RETURN
      END
C
C     SUBROUTINE TO CALCULATE THE AVERAGE CONCENTRATIONS
C     ***********************************************************
      SUBROUTINE AVRAGE(C,CAVRG,N,M,M1)
      REAL C(N,M),CAVRG(N,M1)
      DO 10 I=1,N
      DO 10 J=1,M1
      CAVRG(I,J)=(C(I,J+1)+C(I,J))/2.0
10    CONTINUE
      RETURN
      END
      SUBROUTINE DMAT(MATA,CON,CAT,IN,IK)
C     ***********************************************************
      REAL MATA(IN,IK),CON(IN)
      MATA(1,1)=-CON(1)
      MATA(1,2)=CON(2)
      MATA(1,3)=0.0
      MATA(1,4)=0.0
      MATA(2,1)=CON(1)
      MATA(2,2)=-CON(2)
      MATA(2,3)=-CON(2)
      MATA(2,4)=CON(3)
      MATA(3,1)=0.0
      MATA(3,2)=0.0
      MATA(3,3)=CON(2)
      MATA(3,4)=-CON(3)
      RETURN
      END
      SUBROUTINE MATMT(XX,YY,ZZ,IN,IK)
C     ***********************************************************
      DIMENSION XX(IN,IK),YY(IK),ZZ(IN)
      DO 10 II=1,IN
      ZZ(II)=0.0
      DO 10 JJ=1,IK
      ZZ(II)=ZZ(II)+XX(II,JJ)*YY(JJ)
10    CONTINUE
      RETURN
      END
      SUBROUTINE MATMUT(A,B,C,IK,IJ,IL)
C     ***********************************************************
```

```
      DIMENSION A(IK,IJ),B(IJ,IL),C(IK,IL)
      DO 1 IKK=1,IK
      DO 1 ILL=1,IL
      C(IKK,ILL)=0.0
      DO 1 IJJ=1,IJ
      C(IKK,ILL)=C(IKK,ILL)+A(IKK,IJJ)*B(IJJ,ILL)
1     CONTINUE
      RETURN
      END
      SUBROUTINE TRNMAT(A,AT,IK,IJ)
C     **************************************************************
      DIMENSION A(IK,IJ),AT(IJ,IK)
      DO 1 IJJ=1,IJ
      DO 1 IKK=1,IK
       AT(IJJ,IKK)=A(IKK,IJJ)
1     CONTINUE

      RETURN
      END
C     ****************************************************************
C     PROGRAM SEGMENT 2.:FIRST ALGORITHM
C     ****************************************************************
C     MAIN PROGRAM TO FIND THE OPTIMAL ESTIMATES OF PARAMETERS OF
C          A SET OF ORDINARY DIFFERENTIAL EQUATIONS

C     -- ARRAYS IN COMMON---
      DIMENSION CON(30),CZERO(3),TIME(10)
      COMMON CAT,NVAR,NPARA,NOBS/GUESS/CON,CZERO,TIME
      COMMON KOUNT
      REAL ETA,FSUMSQ,STEPMX,XTOL
      INTEGER I,IFAIL,IPRINT,J,LIW,LJ,LV,LW,M,MAXCAL,L,N,NF,NITER
C     --LOCAL ARRAYS---
      REAL FJAC(30,4),FVEC(30),V(4,4),W(220),S(4),THETA(4)
      INTEGER IW(1)
      EXTERNAL LSQFUN,LSQMON
      DATA CAT,NVAR,NPARA,NOBS,KOUNT/1.0,3,4,10,1/

      N=4
      M=30
      LJ=30
      LV=4
      LIW=1
      LW=220
      READ*,(CZERO(I),I=1,NVAR),(TIME(I),I=1,NOBS),(THETA(I),I=1,NPARA
     1)
      READ*,(CON(I),I=1,30)
      TYPE1,(CZERO(I),I=1,NVAR),(TIME(I),I=1,NOBS),(THETA(I),I=1,NPAR
     1A)
1     FORMAT(1X,'      INPUT:'//6X,'THE INITIAL CONDITIONS C(0) ARE'//
     13(8X,F10.4/)//6X,'THE TIME VECTOR IS'//10(8X,F7.2/)//6X,'THE IN
     1ITIAL ESTIMATES OF PARAMETERS ARE'//(8X,F12.5/)//)
      TYPE2,(CON(I),I=1,30)
2     FORMAT(6X'THE MEASURED VALUES OF THE RESPONSES ARE'//(5X,3F15.5/
     1)//)
      IPRINT=1
      MAXCAL=400*N
      ETA=0.5
      XTOL=1.0E-03
      STEPMX=10.0
      IFAIL=1
```

```
      IFLAG=1
      CALL E04FCF(M,N,LSQFUN,LSQMON,IPRINT,MAXCAL,ETA,XTOL,STEPMX,
     1THETA,FSUMSQ,FVEC,FJAC,LJ,S,V,LV,NITER,NF,IW,LIW,W,LW,IFAIL)
      IF (IFAIL.NE.0)TYPE99998,IFAIL
      IF(IFAIL.EQ.1)GO TO 3
      TYPE99997,FSUMSQ
      TYPE99996,(THETA(J),J=1,NPARA)
      TYPE99995,(FVEC(I),I=1,30)
99995 FORMAT(5X,'VECTOR OF RESIDUALS IS'//(10X,F10.6/)//)
      GO TO  1111
3     TYPE99991
99991 FORMAT(1X,'THE PROGRAM IS NOT CONVERGING')
99996 FORMAT(1X,'         OUTPUT:'//6X,'THE FINAL VALUE OF PARAMETERS IS'/
     1/4(6X,F12.4))
99997 FORMAT(1X,'THE SUM OF SQUARES IS',F12.4)
99998 FORMAT(///16H ERROR EXIT TYPE,I3)
1111  STOP
      END
      SUBROUTINE LSQFUN(IFLAG,M,N,XC,FVECC,IW,LIW,W,LW)
C     **************************************************************
C     ROUTINE TO EVALUATE RESIDUALS
      REAL CCAP(3,10),C1(3),CDT(3),A(3,4),CZ1(3),CZ2(3),CZ3(3)
     1,CCAL(30),K1(3),K2(3),K3(3),K4(3),FVECC(30),W(LW),XC(N),CZR(3)
      INTEGER IW(LIW)
C     SCALAR ARGUMENTS
      INTEGER IFLAG,LIW,LW,M,N
C     ARRAY ARGUMENTS
C     ARRAYS IN COMMON
      REAL CON(30),CZERO(3),TIME(10)
      COMMON CAT,NVAR,NPARA,NOBS/GUESS/CON,CZERO,TIME
      COMMON KOUNT
CC    THE FOLLOWING ARE THE RUNGE-KUTTA GILL CONSTANTS
      DATA AA,B,CC,D/0.2071067,0.2928932,-0.7071067,1.7071067/
      DO 4     IK=1,NVAR
4     CZR(IK)=CZERO(IK)
      DO 100 IN=1,NOBS-1
      DT=TIME(IN+1)-TIME(IN)
      CALL DMAT(A,CZR,CAT,NVAR,NPARA)
      CALL MATMT(A,XC,CDT,NVAR,NPARA)
      DO 5 I=1,NVAR
      K1(I)=DT*CDT(I)
      CZ1(I)=CZR(I)+1.0/2.0*K1(I)
5     CONTINUE
      CALL DMAT(A,CZ1,CAT,NVAR,NPARA)
      CALL MATMT(A,XC,CDT,NVAR,NPARA)
      DO 6 I=1,NVAR
      K2(I)=DT*CDT(I)
      CZ2(I)=CZR(I)+AA*K1(I)+B*K2(I)
6     CONTINUE
      CALL DMAT(A,CZ2,CAT,NVAR,NPARA)
      CALL MATMT(A,XC,CDT,NVAR,NPARA)
      DO 7 I=1,NVAR
      K3(I)=DT*CDT(I)
      CZ3(I)=CZR(I)+CC*K2(I)+D*K3(I)
7     CONTINUE
      CALL DMAT(A,CZ3,CAT,NVAR,NPARA)
      CALL MATMT(A,XC,CDT,NVAR,NPARA)
      DO 8 I=1,NVAR
      K4(I)=DT*CDT(I)
      C1(I)=CZR(I)+1.0/6.0*(K1(I)+K4(I))+1.0/3.0*(B*K2(I)+D*K3(I))
```

```
      CCAP(I,IN)=CZR(I)
      CCAP(I,IN+1)=C1(I)
8     CZR(I)=C1(I)
100   CONTINUE
      TYPE110,((CCAP(I,J),I=1,NVAR),J=1,NOBS)
110   FORMAT(1X,'          OUTPUT'//1X,'THE CONCENTRATION MATRIX IS'//(5X
     13E15.8))
      I=0
      SUMSQ=0.0
      DO 9 K=1,NOBS
      DO 9 J=1,NVAR
      I=I+1
      IF(CCAP(J,K).LT.0.0)CCAP(J,K)=0.0
      CCAL(I)=ABS(CCAP(J,K))
      FVECC(I)=CON(I)-CCAL(I)
9     CONTINUE
      IF(KOUNT.EQ.1)GO TO 11
      RETURN
11    DO 12 I=1,30
12    SUMSQ=SUMSQ+FVECC(I)*FVECC(I)
      TYPE14,(FVECC(I),I=1,30)
      TYPE15,SUMSQ
14    FORMAT(5X,'RESIDUALS IN FIRST ITERATION'//(5X,F10.6/)//)
15    FORMAT(5X,'SUM OF SQUARES IN FIRST ITERATION = ',F15.9)
      KOUNT=KOUNT+1
      RETURN
      END
CCC   SUBROUTINES-----------------------------------------------
      SUBROUTINE MATMT(XX,YY,ZZ,IN,IK)
C     **************************************************************
      DIMENSION XX(IN,IK),YY(IK),ZZ(IN)
      DO 10 II=1,IN
      ZZ(II)=0.0
      DO 10 JJ=1,IK
      ZZ(II)=ZZ(II)+XX(II,JJ)*YY(JJ)
10    CONTINUE
      RETURN
      END
      SUBROUTINE DMAT(A,B,C,IN,IK)
C     *************************************************************
      DIMENSION A(IN,IK),B(IN)
      A(1,1)=-B(1)
      A(1,2)=B(2)
      A(1,3)=0.0
      A(1,4)=0.0
      A(2,1)=B(1)
      A(2,2)=-B(2)
      A(2,3)=-B(2)
      A(2,4)=B(3)
      A(3,1)=0.0
      A(3,2)=0.0
      A(3,3)=B(2)
      A(3,4)=-B(3)
      RETURN
      END
C     DUMMY ROUTINE
      SUBROUTINE LSQMON(M,N,FVECC,FJACC,LJC,S,IGRADE,NITER,NF,IW,LIW
C     ***************************************************************
     1,W,LW)
      TYPE2,NITER,NF
```

```
2     FORMAT(1X,'NITER=',I5,5X,'NF=',I5)
      RETURN
      END
C     **********************************
C     PROGRAM SEGMENT 3. :ALGORITHM 2
C     **********************************
C     PROGRAM TO IMPLEMENT BOX-COMPLEX METHOD TO ESTIMATE PARAMETERS
C     IN MODELS GOVERNED BY DIFFERENTIAL EQUATIONS
      DIMENSION Y(2),KTL(10),SP(7),YL(2),YH(2),CON(4,7),CZERO(4)
     1,TIME(7),C(1)
      COMMON CAT,NN,KK/GUESS/CON,CZERO,TIME
C     ARRAY DATA
      DATA Y(1),Y(2),YL(1),YL(2),YH(1),YH(2)/0.009,0.040,0.007,0.03
     1,0.1,0.1/
      DATA (KTL(I),I=1,10)/5,4,10,3,15,10,500,0,1,0/,(SP(I),I=1,7)/
     11.E-06,1.0E-06,1.5,1.0,0.02,1.0,0.0/
C     SCALAR DATA
      KC=0
      KY=2
      CAT=0.185
      NN=4
      KK=2
      READ*,(CZERO(I),I=1,4),(TIME(I),I=1,7)
      READ*,((CON(I,J),I=1,4),J=1,7)
      CALL CPX(LABEL,KY,YL,YH,KC,KTL,SP,Y,VAL,C,IT)
      TYPE11,VAL
11    FORMAT(5X,'THE OBJECTIVE FUNCTION =',2X,E15.8//)
      TYPE1,(Y(I),I=1,2)
1     FORMAT(5X,'THE  OPTIMUM ESTIMATES OF PARAMETERS ARE'//(5X,F12.4)
     1)
      STOP
      END
      SUBROUTINE VALUE(YT,VALT)
C     ****************************************************
C     THIS SUBROUTINE EVALUATES THE OBJECTIVE FUNCTION REQUIRED
C           FOR MINIMIZATION
C     ****************************************************
      REAL YT(2),CON(4,7),CZERO(4),TIME(7),CCAP(4,7),C1(4),CSTAR(4),
     1CDT(4),A(4,2),CZ1(4),CZ2(4),CZ3(4),K1(4),K2(4),K3(4),K4(4),CCAL
     2(28),FVECC(28),SUMSQ(28),CZR(4),S(4,4),RSD(4,7)
      COMMON CAT,NN,KK/GUESS/CON,CZERO,TIME
CC    THE FOLLOWING ARE THE RUNGE-KUTTA GILL CONSTANTS
      DATA AA,B,CC,D/0.2071067,0.2928932,-0.7071067,1.7071067/
      DO 11    IIK=1,4
11    CZR(IIK)=CZERO(IIK)
      DO 100 N=1,6
      DT=TIME(N+1)-TIME(N)
      CALL DMAT(A,CZR,CAT,NN,KK)
      CALL MATMT(A,YT,CDT,NN,KK)
      DO 2 I=1,4
      K1(I)=DT*CDT(I)
      CZ1(I)=CZR(I)+1.0/2.0*K1(I)
2     CONTINUE
      CALL DMAT(A,CZ1,CAT,NN,KK)
      CALL MATMT(A,YT,CDT,NN,KK)
      DO 4 I=1,4
      K2(I)=DT*CDT(I)
      CZ2(I)=CZR(I)+AA*K1(I)+B*K2(I)
4     CONTINUE
      CALL DMAT(A,CZ2,CAT,NN,KK)
```

```
      CALL MATMT(A,YT,CDT,NN,KK)
      DO 6 I=1,4
      K3(I)=DT*CDT(I)
      CZ3(I)=CZR(I)+CC*K2(I)+D*K3(I)
6     CONTINUE
      CALL DMAT(A,CZ3,CAT,NN,KK)
      CALL MATMT(A,YT,CDT,NN,KK)
      DO 8 I=1,4
      K4(I)=DT*CDT(I)
      C1(I)=CZR(I)+1.0/6.0*(K1(I)+K4(I))+1.0/3.0*(B*K2(I)+D*K3(I))
      CCAP(I,N)=CZR(I)
      CCAP(I,N+1)=C1(I)
8     CZR(I)=C1(I)
100   CONTINUE
      DO 111 I=1,4
      DO 111 K=1,4
      SUM=0.0
      DO 1111 J=1,7
      RSD(K,J)=CON(K,J)-CCAP(K,J)
      TERM=RSD(I,J)*RSD(K,J)
1111  SUM=SUM+TERM
      S(I,K)=SUM
111   CONTINUE
      NORDER=4
      CALL DETER(S,DETS,NORDER)
      VALT=-(10.0**13)*DETS
      RETURN
      END
CCC   SUBROUTINES-----------------------------------------
      SUBROUTINE MATMT(XX,YY,ZZ,IN,IK)
C     ****************************************************
      DIMENSION XX(IN,IK),YY(IK),ZZ(IN)
      DO 10 II=1,IN
      ZZ(II)=0.0
      DO 10 JJ=1,IK
      ZZ(II)=ZZ(II)+XX(II,JJ)*YY(JJ)
10    CONTINUE
      RETURN
      END
      SUBROUTINE DMAT(A,B,C,IN,IK)
C     ************************************************
      DIMENSION A(IN,IK),B(IN)
      TERM=B(2)*C/(2.0*B(1)+B(3))
      TERM1=2.0*B(1)*TERM
      TERM2=B(3)*TERM
      A(1,1)=-TERM1
      A(1,2)=0.0
      A(2,1)=-TERM1
      A(2,2)=-TERM2
      A(3,1)=TERM1
      A(3,2)=-TERM2
      A(4,1)=0.0
      A(4,2)=TERM2
      RETURN
      END
C     DUMMY ROUTINES
C     ***********************************************
      SUBROUTINE TRACE(LAB,ITER)
      RETURN
      END
```

```
      SUBROUTINE NFTRC(LAB,ITER)
      RETURN
      END
      SUBROUTINE INTVAR(YT)
      DIMENSION YT(2)

      RETURN
      END
      SUBROUTINE CNSTRT(CN,JCN)
      DIMENSION CN(JCN)
      RETURN
      END
      SUBROUTINE DETER(A,DTERM,N)
C     ************************************************************
C     PROGRAM TO FIND THE DETERMINANT OF N ORDER MATRIX
      DIMENSION A(N,N)
C     DETERMINANT BY PIVOTAL CONDENSATION
      DTERM=1
      N1=N-1
C     DO ELIMINATION N-1 TIMES
      DO 50 I=1,N1
      II=I+1
C     FIND PIVOTAL ELEMENT
      BIG=0.0
      IROW=I
      ICOL=I
      DO 10 J=1,N
      DO 10 K=1,N
      AB=ABS(A(J,K))
      IF(BIG-AB)2,10,10
2     BIG=AB
      IROW=J
      ICOL=K
10    CONTINUE
      IF(IROW-I)21,21,11
C     EXCHANGE ROW IROW WITH I,
C     MULTIPLY DTERM BY -1
11    DTERM=-DTERM
      DO 20 J=I,N
      TEMP=A(I,J)
      A(I,J)=A(IROW,J)
20    A(IROW,J)=TEMP
21    IF(ICOL-I)31,31,22
C     COLUMN EXCHANGE ICOL WITH I
C     MULTIPLY DTERM BY -1
22    DTERM=-DTERM
      DO 30 J=I,N
      TEMP=A(J,I)
      A(J,I)=A(J,ICOL)
30    A(J,ICOL)=TEMP
31    DTERM=DTERM*A(I,I)
      IF(A(I,I))32,96,32
32    RATIO=1./A(I,I)
      DO 40 K=II,N
C     DIVIDE ROW ELEMENTS BY PIVOT
      A(I,K)=A(I,K)*RATIO
C     ELIMINATE COLUMN ELEMENTS
      DO 40 J=II,N
      TERM=A(J,I)*A(I,K)
      A(J,K)=A(J,K)-TERM
```

```
      DIF=10000.0*ABS(A(J,K))-ABS(TERM)
      IF(DIF)33,33,40
33    A(J,K)=0.0
40    CONTINUE
50    CONTINUE
      DTERM=DTERM*A(N,N)
C     TYPE60,DTERM
60    FORMAT(5X,'DETERMINANT IS'/5X,E15.8)
      RETURN
96    TYPE97
97    FORMAT(10X,13HDETERMINANT=0)
      RETURN
      END
C     ******************************
C     PROGRAM SEGMENT 4. :ALGORITHM 3
C     ******************************
C     MAIN PROGRAM TO IMPLEMENT WEIGHTED RESIDUAL METHOD FOR
C     PARAMETER ESTIMATION IN ORDINARY DIFFERENTIAL EQUATIONS
C     ******************************************************
      REAL C(4,7),CAVRG(4,6),CON(4),DELCON(4),MATA(4,2),MATAT(2,4)
     1,MATATA(2,2),ATDELC(2),SUM1(2,2),SUM2(2),THETA(2),TIME(7),DELT(
     26),DELC(4,6),AA(4,4),WKSPCE(28),BB(4,2),IDEN(2,2),SUM1IN(2,2)
      DATA CAT/0.185/,NN,MM,IA,IB,IC,IAA,IBB/2,2,2,2,2,4,4/,((IDEN(I,J
     1),I=1,2),J=1,2)/1.0,0.0,0.0,1.0/,IFAIL/0/
      READ*,N,M,K
      READ*,((C(I,J),I=1,N),J=1,M),(TIME(I),I=1,M)
      M1=M-1
      CALL AVRAGE(C,CAVRG,N,M,M1)
      CALL DELTA(C,TIME,DELC,DELT,N,M,M1)
      DO 100 II=1,M1
      DO 10 JJ=1,N
10    CON(JJ)=CAVRG(JJ,II)
      CALL DESIGN(CON,MATA,CAT,N,K)
      CALL TRNMAT(MATA,MATAT,N,K)
      CALL MATMUT(MATAT,MATA,MATATA,K,N,K)
      DT=DELT(II)
      DO 20 JJ=1,K
      DO 20 KK=1,K
20    MATATA(JJ,KK)=MATATA(JJ,KK)*DT
      DO 30 JJ=1,N
30    DELCON(JJ)=DELC(JJ,II)
      CALL MATMT(MATAT,DELCON,ATDELC,K,N)
      DO 40 KK=1,K
      SUM1(KK,KK)=SUM1(KK,KK)+MATATA(KK,KK)
      SUM2(KK)=SUM2(KK)+ATDELC(KK)
40    CONTINUE
100   CONTINUE
      CALL F04AEF(SUM1,IA,IDEN,IB,NN,MM,SUM1IN,IC,WKSPCE,AA,IAA,
     1,IFAIL)
      TYPE*,((SUM1IN(I,J),I=1,2),J=1,2)
C     CALL INVRT(SUM1,SUM1IN,K)
      CALL MATMT(SUM1IN,SUM2,THETA,K,K)
      TYPE101,(THETA(I),I=1,K)
101   FORMAT(5X,'THE VECTOR OF PARAMETERS:'//(5X,F8.5/))
      STOP
      END
C     **********************************************************
C     SUBROUTINES START FROM HERE
C     SUBROUTINE TO CALCULATE THE AVERAGE CONCENTRATIONS
      SUBROUTINE AVRAGE(C,CAVRG,N,M,M1)
```

```
      REAL C(N,M),CAVRG(N,M1)
      DO 10 I=1,N
      DO 10 J=1,M1
      CAVRG(I,J)=(C(I,J+1)+C(I,J))/2.0
10    CONTINUE
      RETURN
      END
C     ******************************************************************
C     SUBROUTINE TO CALCULATE THE DIFFERENCE VECTORS
      SUBROUTINE DELTA(C,TIME,DELC,DELT,N,M,M1)
      REAL C(N,M),DELC(N,M1),TIME(M),DELT(M1)
      DO 10 I=1,N
      DO 10 J=1,M1
      DELC(I,J)=(C(I,J+1)-C(I,J))
      DELT(J)=TIME(J+1)-TIME(J)
10    CONTINUE
      RETURN
      END
C     ******************************************************************
C     SUBROUTINE TO EVALUATE THE DESIGN MATRIX 'A'
      SUBROUTINE DESIGN(CON,MATA,CAT,N,K)
      REAL CON(N),MATA(N,K)
      TERM=CON(2)*CAT/(2.0*CON(1)+CON(3))
      X1=2.0*CON(1)*TERM
      X2=CON(3)*TERM
      MATA(1,1)=-X1
      MATA(1,2)=0.0
      MATA(2,1)=-X1
      MATA(2,2)=-X2
      MATA(3,1)=X1
      MATA(3,2)=-X2
      MATA(4,1)=0.0
      MATA(4,2)=X2
      RETURN
      END
C     ******************************************************************
      SUBROUTINE MATMT(XX,YY,ZZ,IN,IK)
      DIMENSION XX(IN,IK),YY(IK),ZZ(IN)
      DO 10 II=1,IN
      ZZ(II)=0.0
      DO 10 JJ=1,IK
      ZZ(II)=ZZ(II)+XX(II,JJ)*YY(JJ)
10    CONTINUE
      RETURN
      END
C     ******************************************************************
      SUBROUTINE MATMUT(A,B,C,IK,IJ,IL)
      DIMENSION A(IK,IJ),B(IJ,IL),C(IK,IL)
      DO 1 IKK=1,IK
      DO 1 ILL=1,IL
      C(IKK,ILL)=0.0
      DO 1 IJJ=1,IJ
      C(IKK,ILL)=C(IKK,ILL)+A(IKK,IJJ)*B(IJJ,ILL)
1     CONTINUE
      RETURN
      END
C     ******************************************************************
      SUBROUTINE TRNMAT(A,AT,IK,IJ)
      DIMENSION A(IK,IJ),AT(IJ,IK)
      DO 1 IJJ=1,IJ
```

```
      DO 1 IKK=1,IK
       AT(IJJ,IKK)=A(IKK,IJJ)
1     CONTINUE
      RETURN
      END
C     *******************************************************
C     THIS SUBROUTINE INVERTS A DIAGONAL MATRIX
      SUBROUTINE INVRT(MMT,MMTIN,K)
      REAL MMT(K,K),MMTIN(K,K)
      DO 10 I=1,K
      DO 10 J=1,K
      IF(I.EQ.J) GO TO 20
      MMTIN(I,J)=0.0
      GO TO 10
20    MMTIN(I,J)=1.0/MMT(I,J)
10    CONTINUE
      RETURN
      END
C     *******************************************************
C     THIS PART OF THE PROGRAM IS TO FIND INTERVAL ESTIMATES
C     *******************************************************
C     THIS PROGRAM FINDS THE INTERVAL ESTIMATES OF A SET OF PARAMETERS
C     EVALUATED FROM A SET OF NONLINEAR O. D .E.'S.
      REAL CZERO(3),CZR(3),C1(3),TIME(10),THETA(4),COVMAT(3,3),
     1GMAT(3,3),AMAT(3,3),BMAT(3,4),DMAT(3,4),DTRANS(4,3),GTRANS(3,3)
     1,SPMAT(4,3),SMAT(3,4),H1MAT(4,4),HMAT(4,4),PMAT(4,4),DPGPM(4,3)
      REAL KSIGMB
C     CAT GIVES THE CATALYST CONCENTRATION USED FOR THE RUN
      COMMON /CONST/CAT,NVAR,NPARA
      DATA CAT,NVAR,NPARA/1.,3,4/
C     ARRAY DATA
C     FOR THIS SET OF ODES.,"G"MATRIX TURNS OUT TO BE IDENTITY MATRIX
      DATA ((DMAT(I,J),J=1,4),I=1,3)/12*0.0/,((GMAT(I,J),J=1,3),I=1,3)
     1/9*0.0/,((HMAT(I,J),J=1,4),I=1,4)/16*0.0/
      READ*,M,L,K
      READ*,(CZERO(I),I=1,3),(TIME(I),I=1,10),(THETA(I),I=1,4)
      READ*,((COVMAT(I,J),J=1,3),I=1,3)
      TYPE1,(CZERO(I),I=1,3),(TIME(I),I=1,10),(THETA(I),I=1,4)
1     FORMAT(5X,'INPUT:'//5X,'THE INITIAL CONDITIONS ARE'//3(5X,F12.4/
     1)//5X,'THE TIME VECTOR IN MINUTES IS'//10(5X,F6.2/)//5X,'THE V
     1ECTOR OF PARAMETERS IS'//(5X,F12.4/)//)
      TYPE2,((COVMAT(I,J),J=1,3),I=1,3)
2     FORMAT(5X'THE ERROR VARIANCE MATRIX IS'//(5X,3F15.5/)//)
      TYPE3
3     FORMAT(5X,'OUTPUT:')
      DO 10 I=1,3
      GMAT(I,I)=1.0
      CZR(I)=CZERO(I)
10    CONTINUE
      CALL TRNMAT(GMAT,GTRANS,M,L)
      DO 100  N=1,9
      DO 25 I=1,3
      DO 25 J=1,3
      DO 25 K=1,4
      AMAT(I,J)=0.0
      BMAT(I,K)=0.0
      DMAT(I,K)=0.0
25    CONTINUE
      DT=TIME(N+1)-TIME(N)
      CALL  RUNGE(CZR,THETA,DT,C1)
```

```
      CALL JACX(C1,THETA,AMAT)
      CALL JACA(C1,THETA,BMAT)
      CALL RKGM(AMAT,BMAT,DT,DMAT)
      CALL TRNMAT(DMAT,DTRANS,M,K)
      CALL MATMUT(DTRANS,GTRANS,SPMAT,K,L,M)
      CALL MATMUT(GMAT,DMAT,SMAT,M,L,K)
      CALL MATMUT(SPMAT,COVMAT,DPGPM,K,M,L)
      CALL MATMUT(DPGPM,SMAT,H1MAT,K,L,K)
      DO 20 I=1,K
      DO 20 J=1,K
      HMAT(I,J)=HMAT(I,J)+H1MAT(I,J)
20    CONTINUE
      DO 30 I=1,3
30    CZR(I)=C1(I)
100   CONTINUE
      CALL INVRT(HMAT,PMAT,K,K)
      DO 40 I=1,4
40    SUM=SUM+PMAT(I,I)
      SIGMAB=SQRT(SUM)
C     THE DEGREES OF FREEDOM ARE = 6.THEREFORE t distribution value
C     TAKEN IS  t=2.571
      KSIGMB=2.571*SIGMAB
      TYPE51
      DO 99 I=1,NPARA
      TYPE50,THETA(I),KSIGMB
50    FORMAT(5X,F12.5,7X,F12.5/)
99    CONTINUE
51    FORMAT(5X,'RELIABILITY OF THE PARAMETERS:'//9X,'PARAMETER',8X,'
     1RELIABILITY'//)
      STOP
      END
C     -----------------------------------------------------------------
      SUBROUTINE RUNGE(CZR,THETA,DT,C1)
C     -----------------------------------------------------------------
      REAL K1(3),K2(3),K3(3),K4(3),CZR(3),C1(3),X(3,4),CDT(3),THETA(4)
     1,CZ1(3),CZ2(3),CZ3(3)
      COMMON /CONST/CAT,NVAR,NPARA
      DATA A,B,C,D/0.2071067,0.2928932,-0.7071067,1.7071067/
      CALL DMAT(X,CZR,CAT,NVAR,NPARA)
      CALL MATMT(X,THETA,CDT,NVAR,NPARA)
      DO 2 I=1,3
      K1(I)=DT*CDT(I)
      CZ1(I)=CZR(I)+0.5*K1(I)
2     CONTINUE
      CALL DMAT(X,CZ1,CAT,NVAR,NPARA)
      CALL MATMT(X,THETA,CDT,NVAR,NPARA)
      DO 4 I=1,3
      K2(I)=DT*CDT(I)
      CZ2(I)=CZR(I)+A*K1(I)+B*K2(I)
4     CONTINUE
      CALL DMAT(X,CZ2,CAT,NVAR,NPARA)
      CALL MATMT(X,THETA,CDT,NVAR,NPARA)
      DO 6 I=1,3
      K3(I)=DT*CDT(I)
      CZ3(I)=CZR(I)+C*K2(I)+D*K3(I)
6     CONTINUE
      CALL DMAT(X,CZ3,CAT,NVAR,NPARA)
      CALL MATMT(X,THETA,CDT,NVAR,NPARA)
      DO 8 I=1,3
      K4(I)=DT*CDT(I)
```

```
      C1(I)=CZR(I)+1.0/6.0*(K1(I)+K4(I))+1.0/3.0*(B*K2(I)+D*K3(I))
8     CONTINUE
      RETURN
      END
CCC   SUBROUTINES----------------------------------------
      SUBROUTINE MATMT(XX,YY,ZZ,IN,IK)
      DIMENSION XX(IN,IK),YY(IK),ZZ(IN)
      DO 10 II=1,IN
      ZZ(II)=0.0
      DO 10 JJ=1,IK
      ZZ(II)=ZZ(II)+XX(II,JJ)*YY(JJ)
10    CONTINUE
      RETURN
      END
      SUBROUTINE DMAT(A,B,C,IN,IK)
      DIMENSION A(IN,IK),B(IN)
      A(1,1)=-B(1)
      A(1,2)=B(2)
      A(1,3)=0.0
      A(1,4)=0.0
      A(2,1)=B(1)
      A(2,2)=-B(2)
      A(2,3)=-B(2)
      A(2,4)=B(3)
      A(3,1)=0.0
      A(3,2)=0.0
      A(3,3)=B(2)
      A(3,4)=-B(3)
      RETURN
      END
C     -------------------------------------------------------
      SUBROUTINE JACX(C1,THETA,AMAT)
C     -------------------------------------------------------
      REAL C1(3),THETA(4),AMAT(3,3)
      COMMON /CONST/CAT,NVAR,NPARA
       AMAT(1,1)=-THETA(1)
       AMAT(1,2)=THETA(2)
       AMAT(1,3)=0.0
       AMAT(2,1)=THETA(1)
       AMAT(2,2)=-(THETA(2)+THETA(3))
       AMAT(2,3)=THETA(4)
       AMAT(3,1)=0.0
       AMAT(3,2)=THETA(3)
       AMAT(3,3)=-THETA(4)
       RETURN
       END
C      ------------------------------------------------------
       SUBROUTINE  JACA(C1,THETA,BMAT)
C      ------------------------------------------------------
       REAL C1(3),THETA(4),BMAT(3,4)
       COMMON /CONST/CAT,NVAR,NPARA
       BMAT(1,1)=-C1(1)
       BMAT(1,2)=C1(2)
       BMAT(1,3)=0.0
       BMAT(1,4)=0.0
       BMAT(2,1)=C1(1)
       BMAT(2,2)=-C1(2)
       BMAT(2,3)=-C1(2)
       BMAT(2,4)=C1(3)
       BMAT(3,1)=0.0
```

```
      BMAT(3,2)=0.0
      BMAT(3,3)=C1(2)
      BMAT(3,4)=-C1(3)
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE RKGM(AMAT,BMAT,DT,DMAT)
C     -----------------------------------------------------------------
      REAL AMAT(3,3),BMAT(3,4),DMAT(3,4),K1(3,4),K2(3,4),K3(3,4),K4(3,
     14),FUNC(3,4),FUNC1(3,4),FUNC2(3,4),FUNC3(3,4),ADMAT(3,4)
      DO 10 I=1,3
      DO 10 J=1,4
      ADMAT(I,J)=0.0
      DO 20 K=1,3
20    ADMAT(I,J)=ADMAT(I,J)+AMAT(I,K)*DMAT(K,J)
      FUNC(I,J)=ADMAT(I,J)+BMAT(I,J)
      K1(I,J)=DT*FUNC(I,J)
      FUNC1(I,J)=FUNC(I,J)+0.5*K1(I,J)
      K2(I,J)=DT*FUNC1(I,J)
      FUNC2(I,J)=FUNC(I,J)+A*K1(I,J)+B*K2(I,J)
      K3(I,J)=DT*FUNC2(I,J)
      FUNC3(I,J)=FUNC(I,J)+C*K2(I,J)+D*K3(I,J)
      K4(I,J)=DT*FUNC3(I,J)
      DMAT(I,J)=FUNC(I,J)+1.0/6.0*(K1(I,J)+K4(I,J))+1.0/3.0*(B*K2(I,J)
     1+D*K3(I,J))
10    CONTINUE
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE MATMUT(A,B,C,IK,IJ,IL)
C     -----------------------------------------------------------------

      DIMENSION A(IK,IJ),B(IJ,IL),C(IK,IL)
      DO 1 IKK=1,IK
      DO 1 ILL=1,IL
      C(IKK,ILL)=0.0
      DO 1 IJJ=1,IJ
      C(IKK,ILL)=C(IKK,ILL)+A(IKK,IJJ)*B(IJJ,ILL)
1     CONTINUE
      RETURN
      END
C     -----------------------------------------------------------------

      SUBROUTINE TRNMAT(A,AT,IK,IJ)
C     -----------------------------------------------------------------
      DIMENSION A(IK,IJ),AT(IJ,IK)
      DO 1 IJJ=1,IJ
      DO 1 IKK=1,IK
       AT(IJJ,IKK)=A(IKK,IJJ)
1     CONTINUE
      RETURN
      END
      SUBROUTINE INVRT(A,AINVRS,MM,KK)
      REAL A(MM,KK),AINVRS(MM,KK)
      DO 10 I=1,MM
      DO 10 J=1,MM
      IF(I.NE.J)GO TO 20
      AINVRS(I,J)=1.0/A(I,J)
      GO TO 10
20    AINVRS(I,J)=0.0
```

```
10    CONTINUE
      RETURN
      END
C     ****************************************************
C     PROGRAM TO PLOT RESIDUALS OBTAINED AFTER ESTIMATION
C     ****************************************************
      DIMENSION TIME(10),Y(10),ISORT(10),CON(4,10)
      READ*,NMODEL
      DO 10 I=1,NMODEL
      READ*,NOBS,NVAR
      READ*,(TIME(J),J=1,NOBS),((CON(J,K),J=1,NVAR),K=1,NOBS)
      DO 30 J=1,NVAR
      DO 20 K=1,NOBS
      Y(K)=CON(J,K)
20    CONTINUE
      NSTEPX=60
      NSTEPY=60
      IFAIL=0
      CALL G01AGF(Y,TIME,NOBS,ISORT,NSTEPX,NSTEPY,IFAIL)
      TYPE*,IFAIL
30    CONTINUE
10    CONTINUE
      STOP
      END
C     *******************************
C     PROGRAM SEGMENT 5. :ALGORITHM 4
C     *******************************
C      PROGRAM TO FIND THE OPTIMAL ESTIMATES OF PARAMETERS OF
C      A SET OF ORDINARY DIFFERENTIAL EQUATIONS USING
C     ONE POINT COLLOCATION METHOD
C     ***********************************************************
C     -- ARRAYS IN COMMON---
      DIMENSION C(28),CZERO(4),TIME(7)
      COMMON CAT,NN,KK/GUESS/C,CZERO,TIME
      REAL ETA,FSUMSQ,STEPMX,XTOL
      INTEGER I,IFAIL,IPRINT,J,LIW,LJ,LV,LW,M,MAXCAL,L,N,NF,NITER
C     --LOCAL ARRAYS---
      REAL FJAC(28,2),FVEC(28),V(2,2),W(125),S(2),X(2)
      INTEGER IW(1)
      EXTERNAL LSQFUN,LSQMON
      DATA CAT,NN,KK/0.185,4,2/
      N=2
      M=28
      LJ=28

      LV=2
      LIW=1
      LW=125
      READ*,(CZERO(I),I=1,4),(TIME(I),I=1,7),(X(I),I=1,2)
      READ*,(C(I),I=1,28)
      TYPE1,(CZERO(I),I=1,4),(TIME(I),I=1,7),(X(I),I=1,2)
1     FORMAT(1X,'      INPUT:'//6X,'THE INITIAL CONDITIONS C(0) ARE'//
     18X,(F10.4/)//6X,'THE TIME VECTOR IS'//8X,(F7.2/)//6X,'THE INITI
     1AL ESTIMATE OF PARAMETERS IS'//8X,(F12.5/)//)
      TYPE2,(C(I),I=1,28)
2     FORMAT(6X'THE MEASURED VALUES OF THE RESPONSES ARE'//8X,(4F15.5/
     1)//)
      IPRINT=1
      MAXCAL=400*N
      ETA=0.5
```

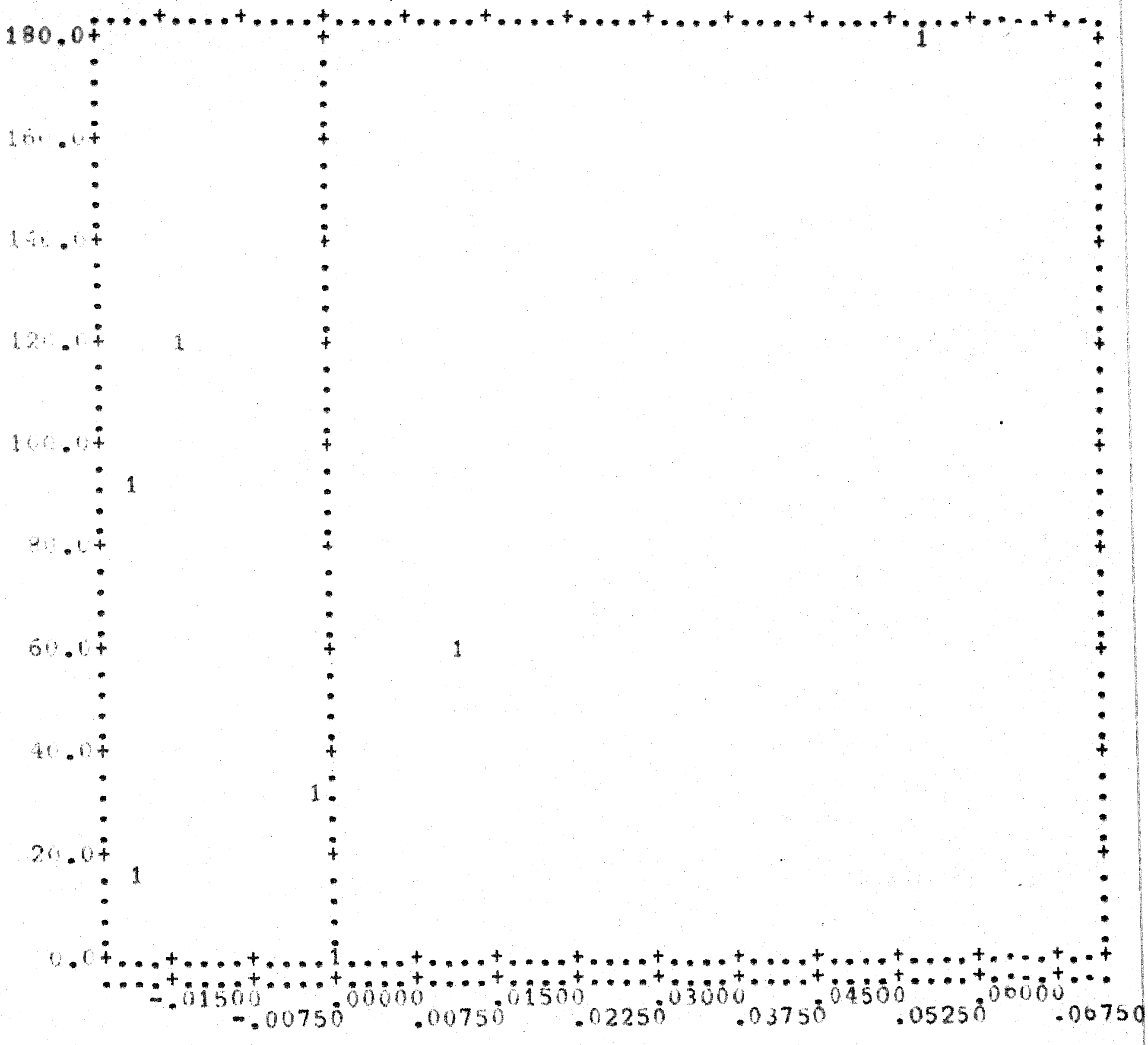
```
      XTOL=1.0E-03
      STEPMX=10.0
      IFAIL=1
      IFLAG=1
      CALL E04FCF(M,N,LSQFUN,LSQMON,IPRINT,MAXCAL,ETA,XTOL,STEPMX,X,
     1FSUMSQ,FVEC,FJAC,LJ,S,V,LV,NITER,NF,IW,LIW,W,LW,IFAIL)
      IF (IFAIL.NE.0)TYPE99998,IFAIL
      IF(IFAIL.EQ.1)GO TO 3
      TYPE99997,FSUMSQ
      TYPE99996,(X(J),J=1,2)
      GO TO  1111
3     TYPE99991
99991 FORMAT(1X,'THE PROGRAM IS NOT CONVERGING')
99996 FORMAT(1X,'      OUTPUT:'//6X,'THE FINAL VALUE OF PARAMETERS IS',
     1/2(6X,F12.4))
99997 FORMAT(1X,'THE SUM OF SQUARES IS',F12.4)
99998 FORMAT(///10H ERROR EXIT TYPE,I3)
1111  STOP
      END
C     **************************************************************
      SUBROUTINE LSQFUN(IFLAG,M,N,XC,FVECC,IW,LIW,W,LW)
C     **************************************************************
C     ROUTINE TO EVALUATE RESIDUALS USING SEMI-IMPLICIT RUNGE-KUTTA
C     METHOD
C     ARRAY ARGUMENTS--
      REAL FVECC(28),W(LW),XC(N)
      INTEGER IW(LIW)
C     SCALAR ARGUMENTS--
      INTEGER IFLAG,LIW,LW,M,N
C     ARRAYS IN COMMON--
      REAL C(28),CZERO(4),TIME(7)
C     LOCAL ARRAYS
      REAL CZR(4),FUN(4),AMAT(4,4),MATRIX(4,4),IDEN(4,4),K1(4),MATINV(
     14,4),F(4),CCAP(4,7),CCAL(28),WKSPCE(10),AA(5,5),BB(5,5),C1(4)
      COMMON CAT,NN,KK/GUESS/C,CZERO,TIME
      DATA ((IDEN(I,J),J=1,4),I=1,4)/16*0.0/
      DO 110 I=1,NN
      CZR(I)=CZERO(I)
110   IDEN(I,I)=1.0
      DO 100 IN=1,6
      DT=TIME(IN+1)-TIME(IN)
      CALL FUNC(XC,CZR,FUN,NN,KK)
      CALL JACX(CZR,XC,AMAT)
      DO 90 I=1,NN
      DO 90 J=1,NN
      MATRIX(I,J)=0.0
      MATRIX(I,J)=IDEN(I,J)-0.5*DT*AMAT(I,J)
90    CONTINUE
      NNN=4
      MMM=4
      IA=4
      IB=4
      IC=4
      IAA=5
      IBB=5
      IFAIL=1

      CALL F04AEF(MATRIX,IA,IDEN,IB,NNN,MMM,MATINV,IC,WKSPCE,AA,IAA,BB
     1,IBB,IFAIL)
      DO 80 I=1,NN
```

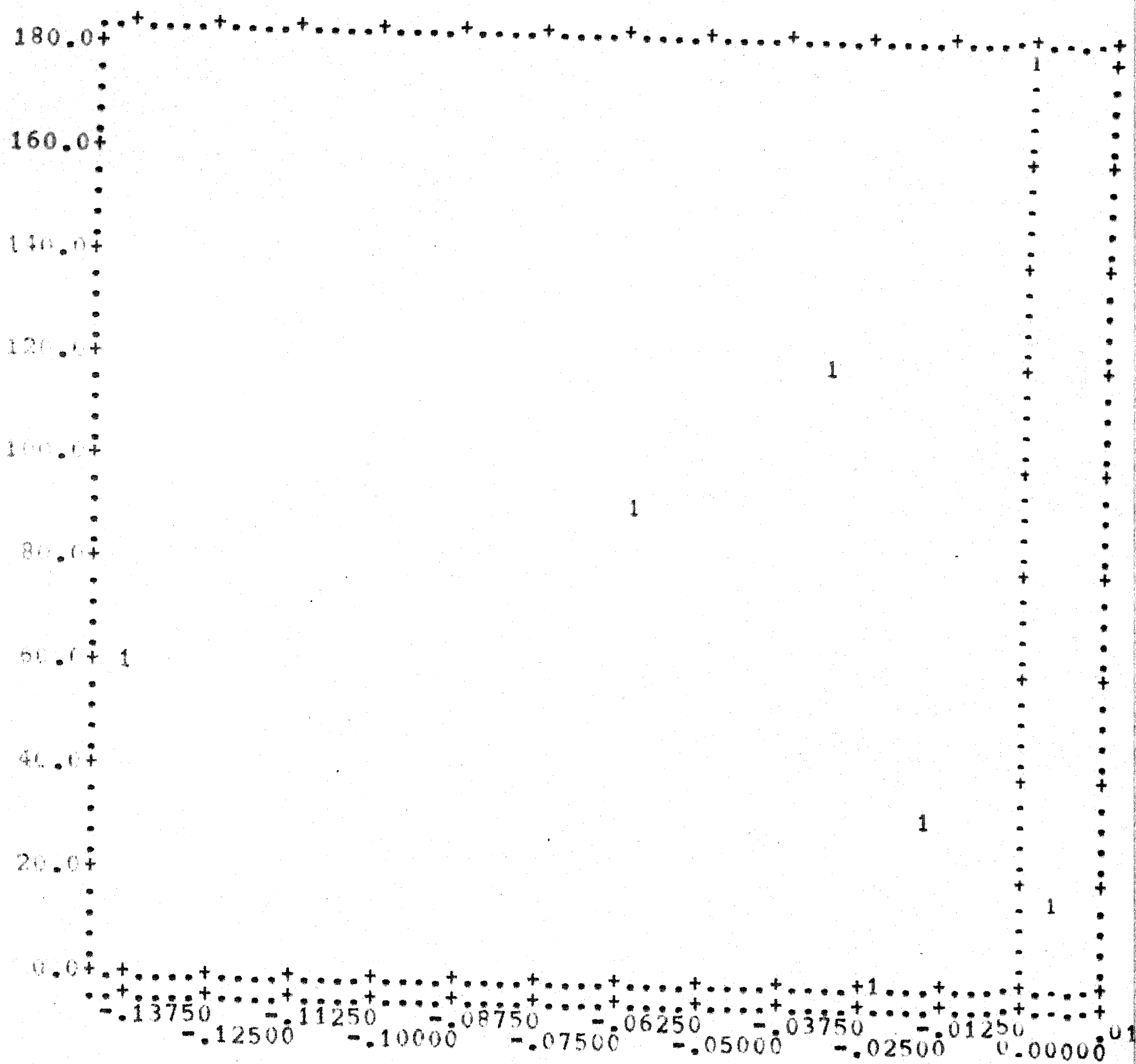
```
80        F(I)=DT*FUN(I)
          CALL MATMT(MATINV,F,K1,NN,NN)
          DO 70 I=1,NN
          C1(I)=CZR(I)+K1(I)
          CCAP(I,IN)=CZR(I)
          CCAP(I,IN+1)=C1(I)
          CZR(I)=C1(I)
70        CONTINUE
100       CONTINUE
          K=0
          DO 60 J=1,7
          DO 60 I=1,NN
          K=K+1
          CCAL(K)=CCAP(I,J)
          FVECC(K)=C(K)-CCAL(K)
60        CONTINUE
          RETURN
          END
CCC       SUBROUTINES-----------------------------------------
          SUBROUTINE MATMT(XX,YY,ZZ,IN,IK)
          DIMENSION XX(IN,IK),YY(IK),ZZ(IN)
          DO 10 II=1,IN
          ZZ(II)=0.0
          DO 10 JJ=1,IK
          ZZ(II)=ZZ(II)+XX(II,JJ)*YY(JJ)
10        CONTINUE
          RETURN
          END
C         DUMMY ROUTINE
          SUBROUTINE LSQMON(M,N,FVECC,FJACC,LJC,S,IGRADE,NITER,NF,IW,LIW
C         ****************************************************************
          1,W,LW)
          TYPE2,NITER,NF
2         FORMAT(1X,'NITER=',I5,5X,'NF=',I5)
          RETURN
          END
          SUBROUTINE FUNC(THETA,CONC,FUN,M,K)
C         ************************************
          REAL CONC(M),FUN(M),THETA(K)
          COMMON CAT,NN,KK
          X1=2.0*THETA(1)*CONC(1)*CONC(2)*CAT/(2.0*CONC(1)+CONC(3))
          X2=THETA(2)*CONC(2)*CONC(3)*CAT/(2.0*CONC(1)+CONC(3))
          FUN(1)=-X1
          FUN(2)=-X1-X2
          FUN(3)=X1-X2
          FUN(4)=X2
          RETURN
          END

          SUBROUTINE JACX(C1,THETA,AMAT)
C         ********************************
          REAL C1(4),THETA(2),AMAT(4,4)
          COMMON CAT,NN,KK
          X1=C1(2)*C1(3)/(2.*C1(1)+C1(3))**2
          X2=C1(1)/(2.*C1(1)+C1(3))
          X3=C1(1)*C1(2)/(2.*C1(1)+C1(3))**2
          X4=C1(3)/(2.*C1(1)+C1(3))
          AMAT(1,1)=(-2.*THETA(1)*CAT*X1)
          AMAT(1,2)=(-2.*THETA(1)*CAT*X2)
          AMAT(1,3)=(2.*THETA(1)*CAT*X3)
```

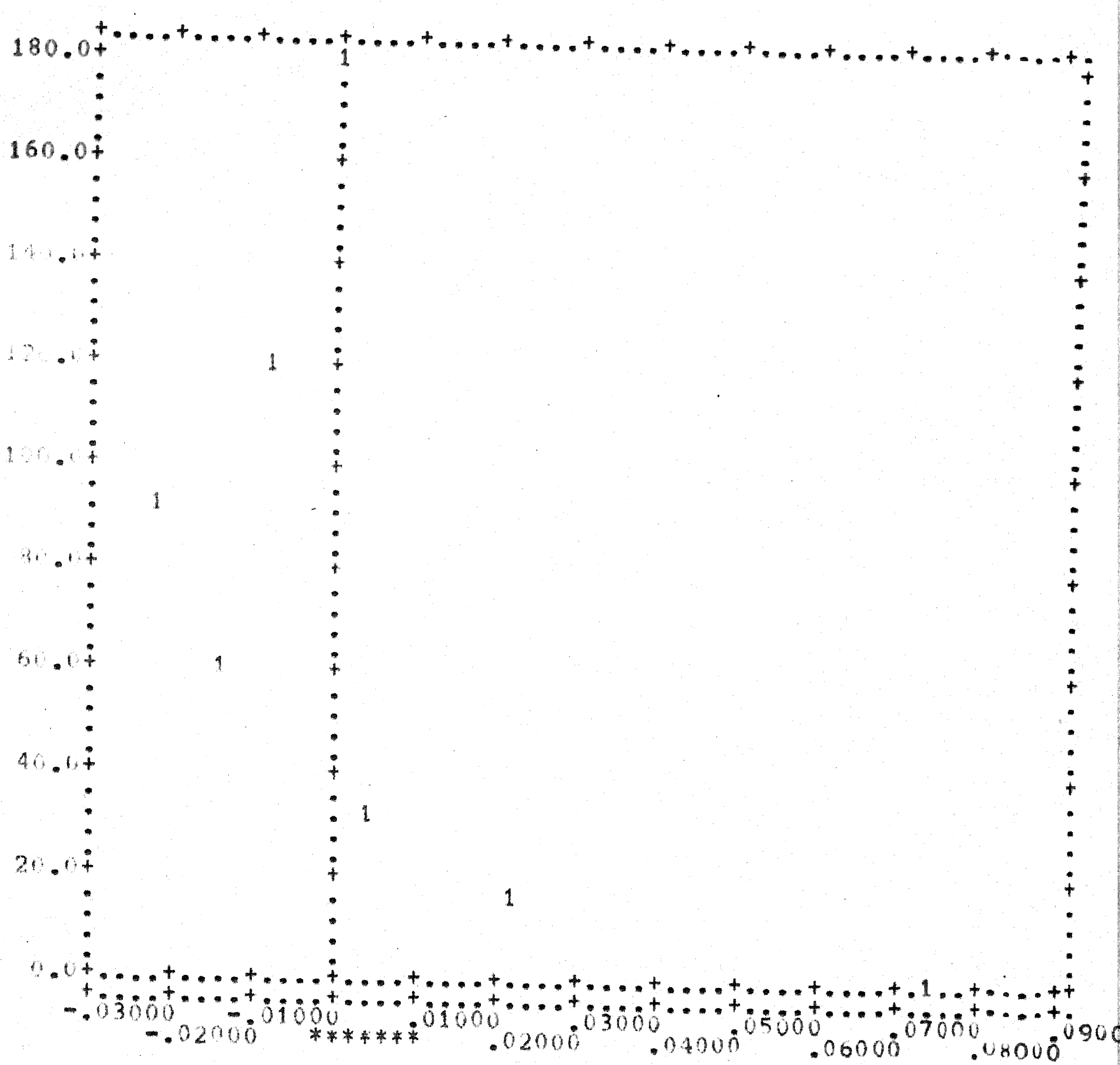
```
      AMAT(1,4)=0.0
      AMAT(2,1)=(-2.0*THETA(1)*CAT*X1+2.*THETA(2)*CAT*X1)
      AMAT(2,2)=(-2.*THETA(1)*CAT*X2-THETA(2)*CAT*X4)
      AMAT(2,3)=(2.*THETA(1)*CAT*X3-2.0*THETA(2)*CAT*X3)
      AMAT(2,4)=0.0
      AMAT(3,1)=(2.*THETA(1)*CAT*X1+2.*THETA(2)*CAT*X1)
      AMAT(3,2)=(2.*THETA(1)*CAT*X2-THETA(2)*CAT*X4)
      AMAT(3,3)=(-2.*THETA(1)*CAT*X3-2.*THETA(2)*CAT*X3)
      AMAT(3,4)=0.0
      AMAT(4,1)=(-2.*THETA(2)*CAT*X1)
      AMAT(4,2)=(THETA(2)*CAT*X4)
      AMAT(4,3)=2.0*THETA(2)*CAT*X3
      AMAT(4,4)=0.0
      RETURN
      END
C     ********************************
      SUBROUTINE INVRT(A,AINVRS,MM,KK)
      REAL A(MM,KK),AINVRS(MM,KK)
      DO 10 I=1,MM
      DO 10 J=1,MM
      IF(I.NE.J)GO TO 20
      AINVRS(I,J)=1.0/A(I,J)
      GO TO 10
20    AINVRS(I,J)=0.0
10    CONTINUE
      RETURN
      END
         +....+....+....+....+....+....+....+....+....+....+....+....+...
  240.0+                                                +
         .                                              .
         .                                              .
         .                                              .
         .                                              .
    2
```

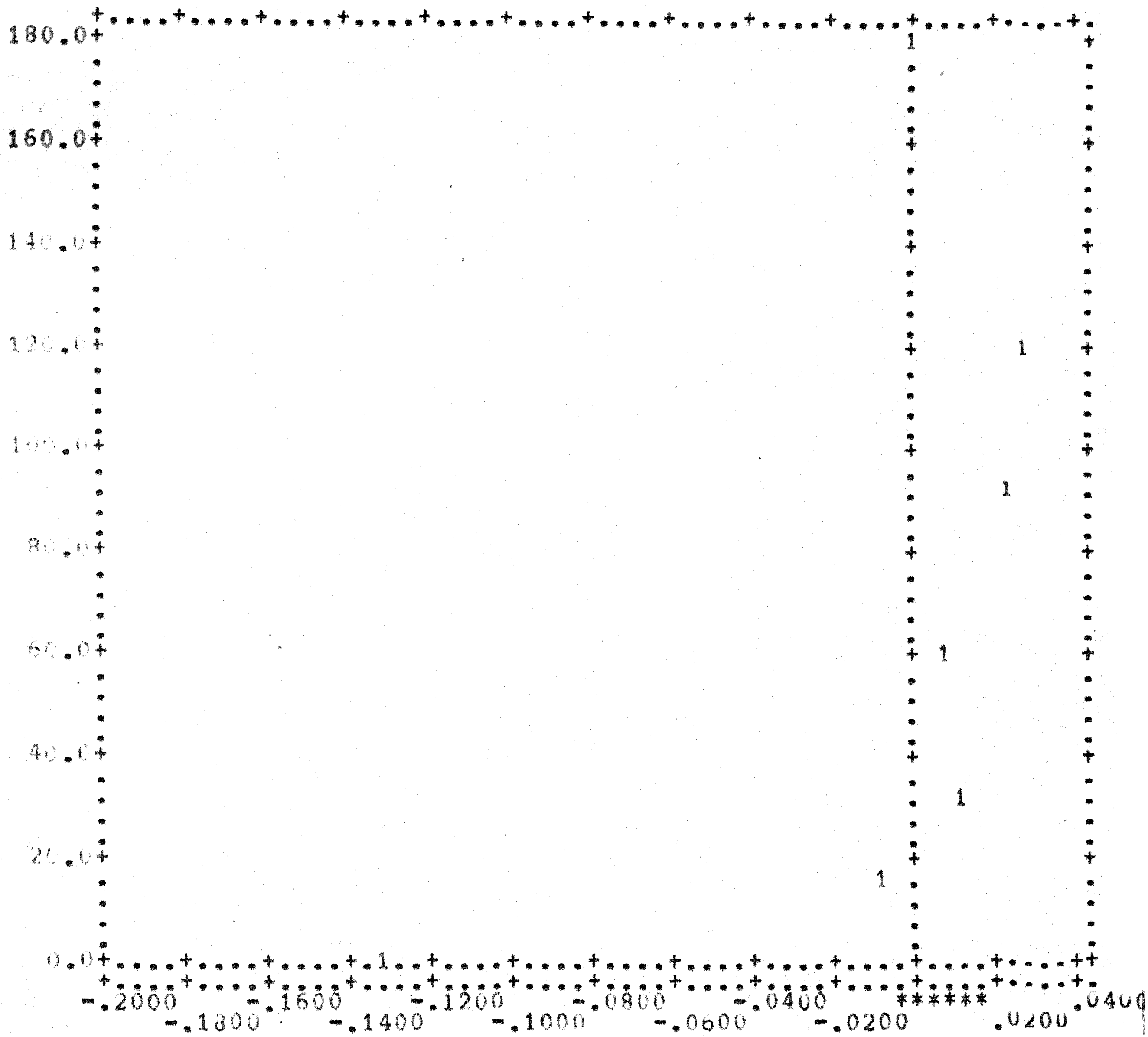
MODEL1
****

TIME SEQUENCE PLOTS OF RESIDUALS
****

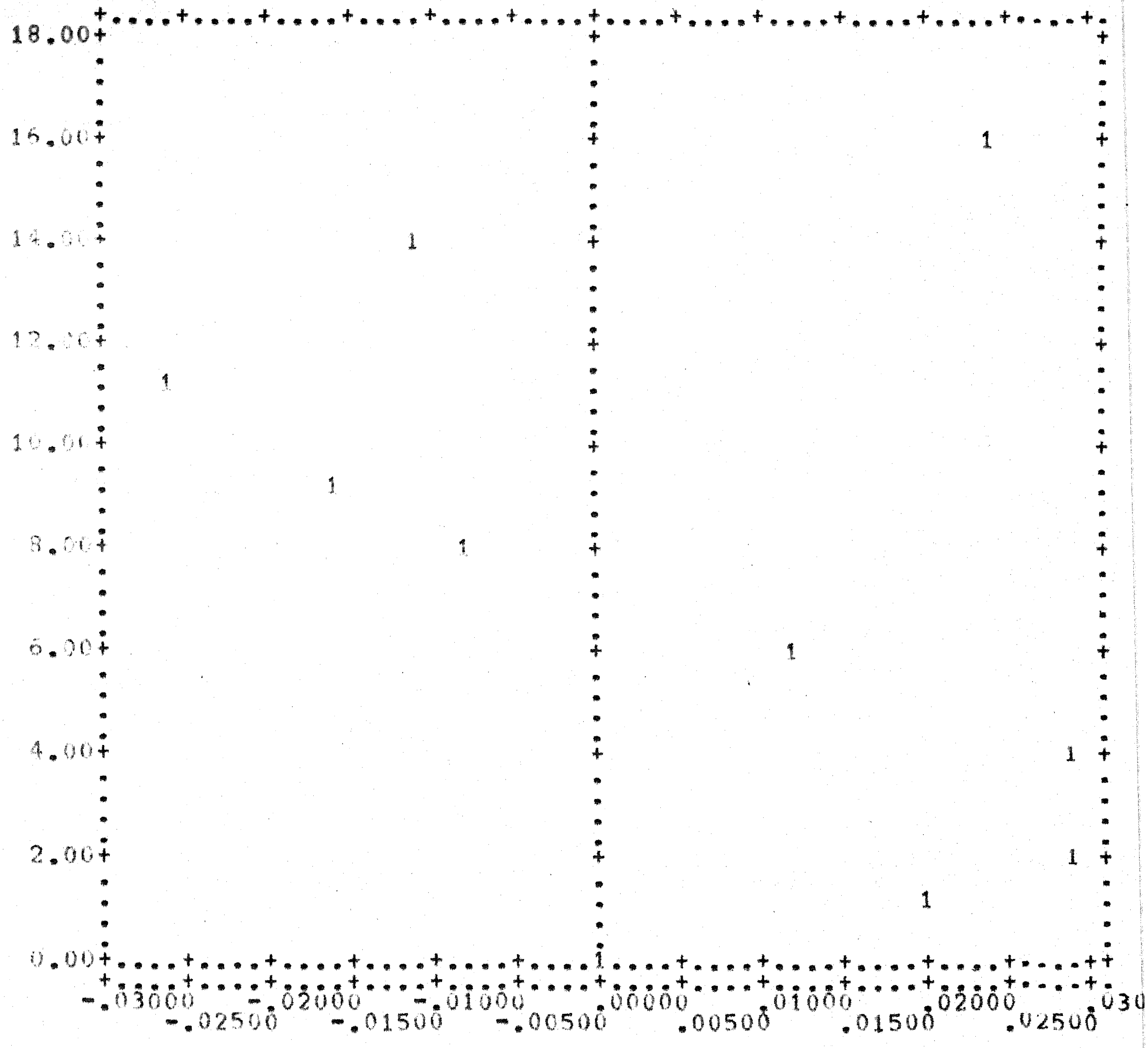


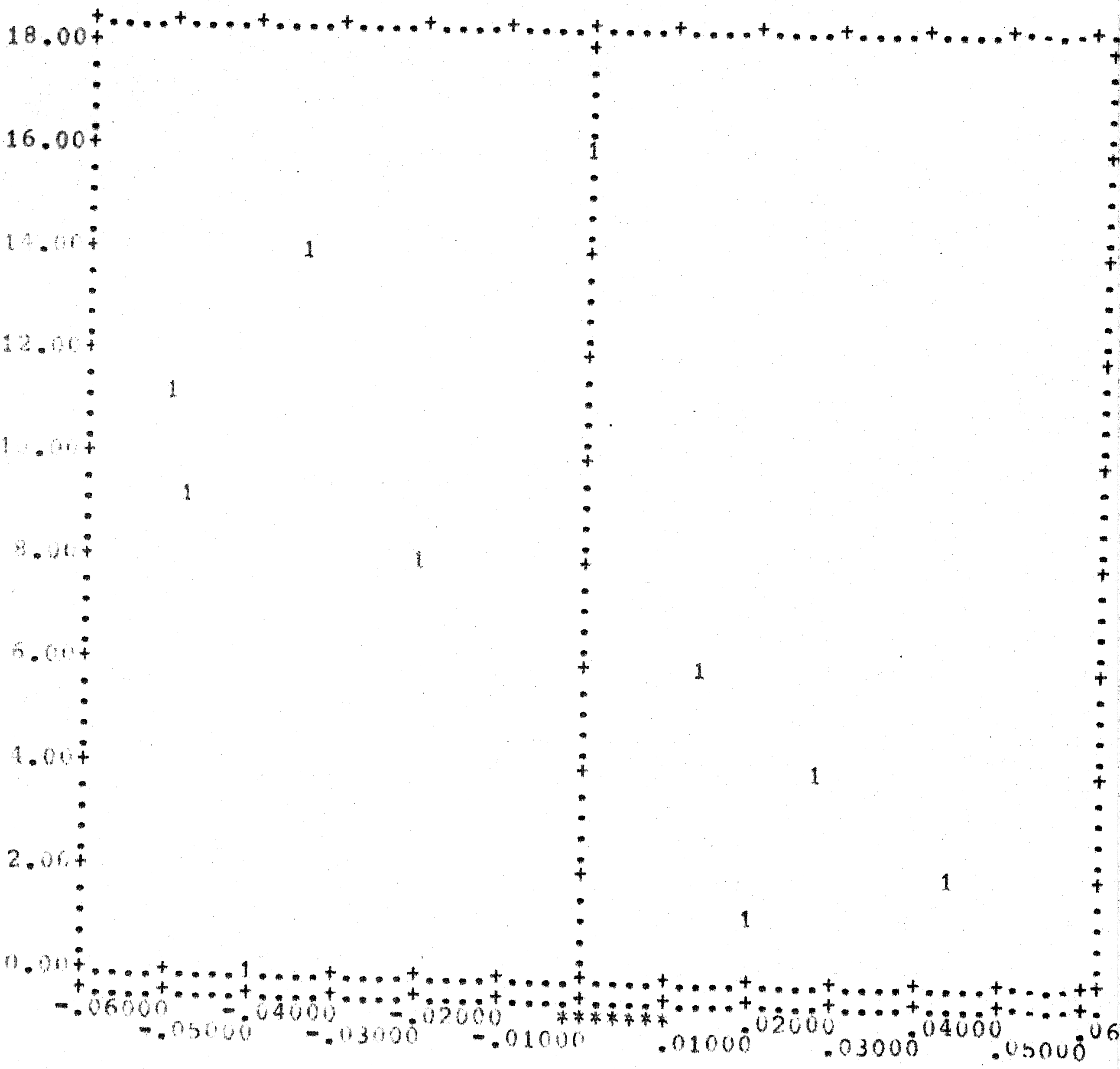

180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-.13750
-.11250
-.08750
-.06250
-.03750
-.01250
-.12500
-.10000
-.07500
-.05000
-.02500
0.00000

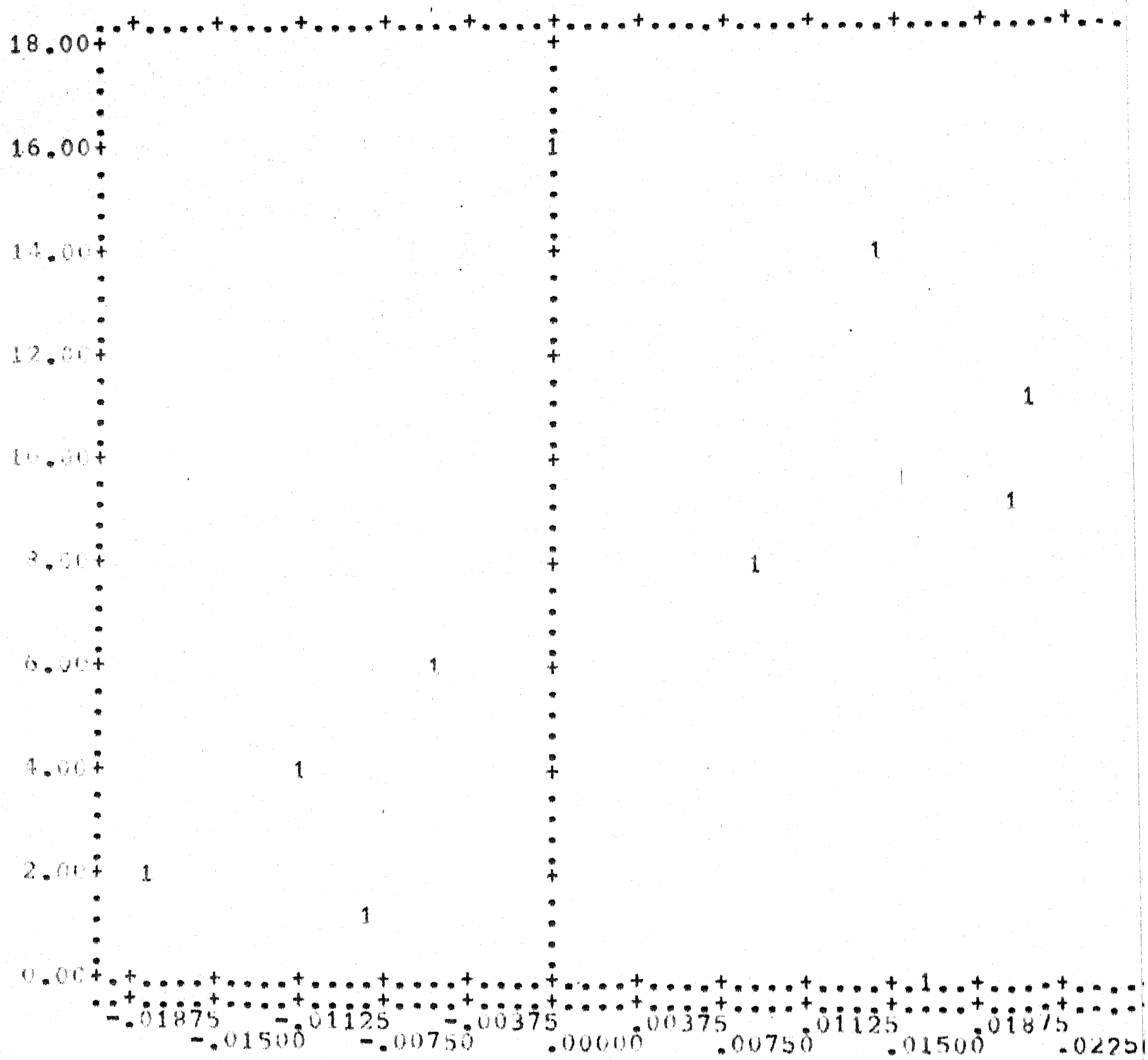

180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-.03000
-.02000
-.01000
****
.01000
.02000
.03000
.04000
.05000
.06000
.07000
.08000
.0900


180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-.2000
-.1800
-.1600
-.1400
-.1200
-.1000
-.0800
-.0600
-.0400
-.0200
.0200
.0400

MODEL2
****
RESIDUAL PLOTS FOR MODEL2
****




18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-.06000
-.05000
-.04000
-.03000
-.02000
-.01000
.01000
.02000
.03000
.04000
.05000
.06


18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-.01875
-.01500
-.01125
-.00750
-.00375
.00000
.00375
.00750
.01125
.01500
.01875
.0225